DOCUMENT RESUME

ED 119 767 JC 760 160

AUTHOR Howard, Alan; And Others
TITLE A Plan for Community College Instructional Computing.
INSTITUTION Washington Association of Community Colleges.; Washington State Board for Community Coll. Education, Olympia.
PUB DATE Dec 75
NOTE 82p.

EDRS PRICE MF-$0.83 HC-$4.67 Plus Postage
DESCRIPTORS Community Colleges; Computer Assisted Instruction; *Computers; Computer Science; *Computer Science Education; Course Descriptions; Course Objectives; Curriculum Development; Curriculum Guides; *Data Processing; Equipment Utilization; *Junior Colleges; *Statewide Planning
IDENTIFIERS *Washington

ABSTRACT

This document presents a comprehensive plan for future growth in instructional computing in the Washington community colleges. Two chapters define the curriculum objectives and content recommended for instructional courses in the community colleges which require access to computing facilities. The courses described include data processing technology, problem solving, general introduction to computers and computing, and computer science. The use of computers as adjuncts to classes is considered in sections on computer assisted instruction, computer based instruction, and computer managed instruction. Other chapters define the software, hardware, peopleware and other support required to adequately meet the requirements of the identified course contents and objectives. The final chapter presents the resource specification and acquisition procedures which will permit the community colleges to move from the present level of fragmented, individualized approaches to meeting the instructional computing needs of students, to a planned and coordinated distributed computing network of compatable hardware and software, and to a more efficient utilization of total state computing resources. A glossary and bibliography are appended. (Author/DC)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED119767

U S DEPARTMENT OF HEALTH
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

A PLAN FOR COMMUNITY COLLEGE

INSTRUCTIONAL COMPUTING

WISC TASK IV.III

DECEMBER, 1975

JC 760 160

STATE OF WASHINGTON
State Board for Community College Education

Forward

The Washington Association of Community Colleges (WACC) in February, 1974, established a fourth commission, WISC (WACC Information Systems Commission) in order to plan, develop, and implement a comprehensive and consolidated solution to the use of computers in the community college system of Washington State.

Pursuant to its charter, WISC (WACC Information Systems Commission) has identified a number of task groups associated with the use of data processing and computing activities in the community college system.

In the area of instructional computing, WISC designated in July, 1974, two task groups, TASK III, Academic Instructional Computing, and TASK IV, Occupational Instructional Computing, charged with the development of instructional computing plans for their respective areas. In April, 1975, TASK III was combined with TASK IV in order to facilitate the development of a combined plan for instructional computing. The members of the combined group, entitled TASK IV.III, who participated in the development of the plan are:

Mr. Elton Chase, WISC
Liaison Representative
Clark College

Ms. Sherie Cox
State Board for Community
College Education

Ms. Roberta Giovannini
Data Processing Authority

Mr. Jerry Hill
Bellevue Community College

Mr. Howard Hubbard
Shoreline Community College

Mr. Gary Isham
Edmonds Community College

Mr. Ken Michelsen
Highline Community College

Mr. George Miller
North Seattle Community College

Mr. Douglas Nielsen
Yakima Valley College

Ms. Dee Stark
Seattle Central Community College

Dr. Alan Howard, Chairman
Lower Columbia College

In accordance with the WISC charter, TASK IV.III assumed the following duties:

1) Research and documentation of instructional computing curriculum objectives and curriculum content.

2) Research and documentation of software, hardware, and peopleware resources and configurations necessary to

support the identified instructional computing curriculum objectives and content.

3) Development of resource acquisition procedures, schedules, and approval mechanisms for instructional computing.

4) Development of recommendations for implementation and maintenance of the Plan for Instructional Computing.

A subsequent planning session held in September, 1975, produced a "conceptual plan" which provided much of the framework for the development of this Plan for Instructional Computing. The conceptual plan developed defined a "stratified delivery structure" of computing resources for the community college system and established the need for an organizational structure within which to implement the plan.

In the development of this plan, several assumptions regarding host computing support of community college computing needs, state-of-the-art of computing technology, and the direction of community college computing were made in order to form a basis from which to begin. It was assumed that the primary mode of computing support for the community colleges will be a distributed computing network, that adequate external, remote host facilities will be provided by identified state host centers, and that technological developments in computing hardware and software will continue to provide an expanding set of alternatives for delivery of instructional computing resources to the community college campuses of Washington State.

Although this plan is expected to make a significant contribution to the community college computing environments of this state in itself, it is also expected that an organizational structure within which to implement, maintain, and update the plan will provide an optimal means to coordinate instructional computing across the system in an orderly and efficient manner. Therefore, it is recommended that a WACC subcouncil, comprised of instructors and administrators, WICCC (WACC Instructional Computing Coordinating Council) be formed to serve as a monitoring group in the area of Instructional Computing. The council would review and make recommendations in the areas of software, hardware, delivery systems, and revisions to the Plan for Instructional Computing. The council formed would have as objectives the following:

1. To maintain a current profile of system-wide instructional support and services which involve the use of a computer.

2. To recommend pilot projects involving the use of computers which will enhance the instructional programs.

3. To exchange with all districts the experiences gained by each district in using the computer as a teaching aid via

publication of a newsletter or some other information medium.

4. To assemble a library of CAI (Computer-Assisted Instruction), CMI (Computer-Managed Instruction), CBI (Computer-Based Instruction) and problem-solving applications and documentation in use in the community college system.

5. To develop or adopt at least one each Computer-Assisted, Computer-Managed, and Computer-Based instruction module for demonstration purposes throughout the system.

6. To disseminate information on cost-effective computing techniques, including cost comparisons for different computer utilities, terminals, applications and use techniques.

7. To establish a framework whereby future developments can be introduced, discussed, and accepted into the existing system.

It is also suggested that a source of funding be sought for a one-year period to support a WICCC Coordinator for the purposes of providing a transition from the present status to that provided by the Plan, organizing the WICCC structure, and developing the necessary guidelines, data sets, and reporting formats to be used by WICCC.

Finally, it is the wishes of TASK IV.III to express appreciation to those members of the community college system, the State Board for Community College Education, the Data Processing Authority, WISC, WACC, and others who aided and assisted the committee in their task. Without the support of these individuals and agencies the Plan would not be possible.

Table of Contents

		Page
	Preface	
I	Introduction	
II	Curriculum Objectives	12
	II.1 Overview	12
	II.2 Data Processing Technology	12
	II.3 Problem Solving	15
	II.4 General Introduction to Computer & Computing	17
	II.5 Computer-based Instruction	18
	II.6 Computer-Managed Instruction	19
	II.7 Computer-Assisted Instruction	20
	II.8 Computer Science Major (Academic)	21
III	Curriculum Content--Typical	22
	III.1 Overview	22
	III.2 Data Processing Technology	22
	III.3 Problem Solving Courses	31
	III.4 General Introduction Courses	36
	III.5 Computer-based Instruction Courses	39
	III.6 Computer-managed Instruction	39
	III.7 Computer-assisted Instruction	40
	III.8 Computer Science Major	42
IV	Software Support Required	44
	IV.1 Overview	44
	IV.2 Data Processing Technology	44
	IV.3 Academic Instructional Computing	47
V	Hardware Support Required	49
	V.1 Overview	49
	V.2 Data Processing Technology	49
	V.3 Academic Instructional Computing	50
	V.4 Stratified Delivery System	52
	V.5 Detail Levels, Stratified Delivery System	53
VI	Peopleware Support Required	59
	VI.1 Overview	59
	VI.2 Data Processing Technology	59
	VI.3 Academic Instructional Computing	60

VII Other Requirements 62
VII.1 Introduction 62
VII.2 Personnel Development 62
VII.3 Instructional Aids 64
VII.4 Facilities 64

VIII Resource Specifications and Acquisition Procedures 65
VIII.1 Introduction 65
VIII.2 Acquisition Procedures - Hardware 65
VIII.3 Software Acquisition & Maintenance 66
VIII.4 Acquisition Procedures - Peopleware 67
VIII.5 Criteria for Movement onto and within the Stratified Delivery Structure 67

IX Appendices
Appendix A - Resource Acquisition Forms and Documents 70
Appendix B - Glossary of Terms 75

X Bibliography 78

I INTRODUCTION

The importance of the computer in today's community college cannot be overemphasized. The scope of its use includes every facet of operation from teaching and learning through administrative operations and community service.

The report on Computers in Higher Education, prepared by the President's Science Advisory Committee, emphasizes the role of the computer in the following excerpts:

(1) In all parts of education, government, or industry, digital computer use has come about because it is an effective tool.

(2) In attempting to assess the educational need for computers in colleges and universities, we find ourselves compelled to believe that by 1978 essentially all university and college students will require some basic understanding of digital computation.

(3) Every such institution will require on its staff enough faculty with computer experience to teach computer use and provide computer experience in various disciplines.

The use of computers in higher education can be dichotomized into two categories: use of the computer as a tool, and, study of the computer as a subject in its own right. In the first area, as a tool, the computer is identified with "academic" use as a problem solving aid, as a vehicle for simulation modeling and gaming, as a manager of learning, and as a "main line" and "adjunctive" aid in instruction for any activity.

As subject matter the computer is identified with the vocationally oriented data processing technology curriculum and the academic computer science major.

The community colleges in the State of Washington were among the first colleges in the nation to develop educational programs in Data Processing Technology. These programs were established in 1962 through the combined efforts of the Division of Vocational Education, the colleges (which were vocational-technical schools and junior colleges at that time), and interested vendors. A recommended curriculum was developed and unit record equipment and IBM 1620 computers were installed in these schools during 1962-63. Funding for these instructional programs came primarily from the local school districts with assistance from vendors, in the form of a 60% educational discount on hardware, and from supplemental federal vocational education funds.

Vocational training programs were established and developed through the years by a number of colleges with the assistance and

advice of local advisory committees representing data processing installations from local industry. Over the past twelve years, each of the fifteen approved vocational training programs has evolved by adapting its curriculum and hardware resources to meet the needs of the local communities. Therefore, while the same general subject areas are covered, each district currently offers a unique sequence of courses and instructional emphasis which varies by geographical area. For example: Columbia Basin Community College, in Pasco, is strongly influenced by the emphasis on scientific computing by local industry in the Tri-Cities area; Olympic College, in Bremerton, conducts an extensive evening program to meet the needs of employees of the various U.S. Naval installations; the instructional program at Centralia College is strongly oriented toward the state government employment market in Olympia; and, Clark College in Vancouver, Washington, trains students for employment in the Portland Metropolitan Area.

Evolving from the vocational programs, generally, academic computing relied initially on the hardware, software, and people ware associated with the vocational programs. With the growth and accessibility of commercial time sharing services and four-year college/university facilities, the community colleges began to expand problem solving, CBI, and general computing courses in their curricula about 1968. Utilizing a diverse collection of software and hardware resources, academic computing applications in the colleges now ranges from none to extensive. Some colleges, such as Lower Columbia College in Longview, have Chemistry CAI applications, Business and Management simulations, mathematics and Statistics CMI systems, as well as introductory and programming courses. Others in the system are currently "have-nots" as computing is relatively non-existant on some campuses.

Hardware support, also, for the instructional programs is as varied and mixed as the programs being offered.

Many instructional programs, due to the obsolescence of their in-house hardware, have relied on purchased or donated time from service bureaus or private companies and other schools to provide students access to modern equipment.

Having as a requirement the provision of adequate education to those students enrolled in its institutions, the State of Washington has a specific obligation to provide adequate computing support for these students. Adequate computing support is directly related to skill requirements graduates must have to fill jobs in the late 70's and the 80's. With the rapid increase in the numbers and uses of computers in society, there is pressure upon both the student and the institutions of higher learning to increase their contact with computer resources and delivery mechanisms.

A few years ago, highly intelligent, well-trained technicians were about the only persons able to communicate effectively with

computers. Today, computer systems are designed for use by persons from all walks of life, from all levels of intelligence and training. The rate of computerization of society is accelerating. The impact of this acceleration is having significant effects on the way of life in society.

By the year 1985, according to Martin and Norman (1970) in The Computerized Society, "there will be a sudden, massive spread of computer usage that will affect the lives of almost everyone." Attributing this spread to four factors:

(1) mass production of computer hardware at significantly lower cost

(2) standardization of software and application systems

(3) large data bases and rapid access storage devices

(4) telecommunication network extensions and expansions

these authors further contend that rapid changes in information technology open a wide range of possible futures for society.

Education must strive to increase knowledge and understanding of computers by society. Community colleges, having the greatest contact with the members of society, being the "educational institutions of the masses", have the greatest opportunity to provide the necessary learning experiences for the greatest number of people. No other institution has greater potential for developing the requisite understanding and knowledge to manage and plan the use of information technology for the betterment of society.

Until recently, community colleges of this state responded to the increased demand for computing education much in the same manner that other colleges across the nation responded--they sought to improve their condition on an individualized basis to the best of their abilities. However, the development and implementation of computer networks have significantly expanded the alternatives facing the individual college in its search for methods to improve the quality and quantity of computing to its students at minimum cost. Technically feasible for several years, with attractive cost/benefit attributes, the network concept is revolutionizing college computing, according to G. P. Weeg in an address to the Pacific Northwest Cooperative Computing Center Second Annual Symposium held April 13, 1973 at Washington State University. Pointing out that networks exist in about 20 locations in the nation, most of them initiated under the auspices of NSF funding, Weeg states that these networks provide to smaller institutions computing support of a kind more potent that any small college could in any other way achieve. Since it appears that continuation of the trend toward development of individual computer centers to support

the growing computing needs of each campus would exceed the capability of the funding source to support such ventures, it appears that the community colleges of this state should steer a course of future development that is consistent with the utilization of computer networks and distributed computing resources whenever practical.

Distributed computing systems are those systems which provide diverse and appropriate computer capability at geographically separated locations. Usually a computer network is involved in the distribution of such resources to remote users. A network consists, typically, of a central computer facility servicing remotely located terminals via a communications facility. Distributed computing systems may include networks, may include special-purpose dedicated computing facilities, or may include a combination of both network and dedicated facilities.

A number of educational computing network-based computer systems are currently reevaluating their systems relative to the impact of technological advances in computer terminals and minicomputers. The University of Iowa, for example, finding only 20% of its classes making use of the computer for instructional purposes (compared to 88% of the student-body at Dartmouth College) concluded that RJE batch systems were insufficient tools in terms of supporting the developing needs for computing service within the institution. Iowa further concluded that time-sharing conversational computing facilities, free to the student, employing BASIC was necessary to enhance and improve the level of computing at their institution. It is interesting to note that the Iowa study, reported in ON-LINE, March, 1974, found that one computer terminal can optimally serve about 40 students who receive one computing assignment per week. Aiming at a 50% level of use by the student-body, Iowa observed that it was economically and technologically feasible to install mini-computers interfaced to the central computer facility to provide the necessary computing support to reach the 50% figure. The minis at Iowa are connected to the central computer for batch RJE support. The minis are also used for access to the time-sharing capabilities of the host that exceed the capabilities of the mini; however, the minis are used to meet the bulk of time-sharing conversational computing demands.

The University of California at San Francisco has used computers in instruction for over 12 years. Initially utilizing a second generation computer dedicated to a CAI system called COMPTEST in 1962, a new system called PILOT was implemented when a large, centralized time-sharing facility became available. Recent advances in computer technology in the form of dedicated, stand-alone, programmable computer terminals now allow UCSF to utilize "intelligent" programmable terminals off-line for special purpose CAI applications more efficiently and economically than on-line to the central computer, plus provides on-line capability for time-sharing on the central computer when necessary and appropriate.

The developments in the technology of programmable terminals and minicomputers allows the University of Iowa and the University of California at San Francisco to deliver necessary and appropriate computer support in a distributive fashion via centralized and remote facilities in an economical and efficient manner.

From the perspective of the small college, the distributed computing concept offers the potential to move from no computing to a fairly sophisticated level in a matter of a year or two, and by furthering cooperation between institutions, the distributed computing network gives the community colleges of Washington State viable means of providing necessary and appropriate computing services to all in an economical and efficient manner.

This document presents a comprehensive plan for future growth in instructional computing in the community colleges in the State of Washington. The following two chapters define the curriculum objectives and content required for instructional courses in the community colleges which require access to computing facilities. Chapters 4, 5, 6 and 7 define the software, hardware, peopleware and other support required to adequately meet the identified course content and objectives. Chapter 8 presents the resource specifications and acquisition procedures which will permit the Colleges to move from the present level of fragmented, individualized approaches to meeting the instructional computing needs of students, to a planned and coordinated distributed computing system of compatable hardware, common software, and more efficient utilization of total State computing resources.

II CURRICULUM OBJECTIVES

II.1 OVERVIEW

The community college curricular offerings in the area of instructional computing have as general objectives the following:

1) provide professional EDP career development programs so that graduates can be successfully placed in EDP entry-level positions

2) Furnish additional computer related skills so that EDP graduates may be more competitive in their job seeking

3) acquaint students with aspects of computing sufficient to provide an understanding of the role, use, limitations, and potential in society

4) Provide auxilliary computing courses sufficient to fulfill the need for problem solving by computer for all students

5) provide computer-based, computer-assisted, and computer-managed support in order to enhance traditional course objectives.

The following sections describe in detail the curriculum and application objectives for each of the seven areas of community college instructional computing.

II.2 DATA PROCESSING TECHNOLOGY

The students currently being served by the data processing technology programs in the Community college system can be divided into two distinct groups. First, there are those whose objective is to obtain a data processing position in industry and second, there are the students who begin in data processing, but either leave school or transfer to other majors.

The first group is comprised of data processing majors who will use their training in data processing positions in industry. They can be classified into three groups: a) data entry operators, b) digital computer operators, and c) business computer programmers. These students spend a high percentage of their class hours in data processing classes with additional training added to improve their employability.

The second group is made up primarily of former data processing majors who have either changed majors after beginning in data processing, or who have left school, but who are also more employable because of their data processing training. Curriculum for these people will be the same as for the data processing major for as long as they are in the program. It is important that

beginning courses be arranged so that these students gain as much skill as possible during their time in the data processing program.

II.2.1 Data Processing Majors

Most of the objectives and training requirements for this group of students has been extracted from the Occupational Analysis for: Business Data Processing Technology published June, 1972, by the Coordinating Council for Occupational Education of the State of Washington. These have been expanded based on current technology.

Data Entry Operator

1. The ability to communicate effectively with other data processing personnel.
2. The ability to prepare program cards for IBM 029 type machines.
3. The ability to prepare a program for a buffered data entry system.
4. The ability to operate both buffered and non-buffered data entry machines.
5. An understanding of a data processing system's "life cycle", i.e. the sequence of data flow through the data processing system.
6. An understanding of computer hardware, software, and operating environment in order to perform effectively as a data entry operator.
7. Ability to operate non-card data entry equipment (i.e. key-to-tape and key-disk).
8. Experience in the operations of on-line data entry equipment.

Digital Computer Operator

1. A comprehensive understanding of computer hardware, software, and operating environment in order to perform effectively.
2. An understanding of a data processing system's "life cycle."
3. At least 300 hours of hands on experience prior to graduation.
4. Hands-on training on a state-of-the-art computer systems.

5. An understanding of mini computer systems.

6. Hands-on training on mini computer and small business systems.

Computer Programmer (Two-year Degree)

1. The ability to program effectively in a high-level business language, preferably ANSI COBOL.

2. The ability to perform basic computer operations functions.

3. Experience in assembly language coding;

4. A comprehensive understanding of computer hardware, software, and operating environment in order to perform effectively as a business programmer.

5. An understanding of the capabilities of large and small computing systems, e.g., multiprocessing and multiprogramming.

6. Basic data communication concepts and techniques, such as remote batch, time-sharing, direct access, and message switching.

7. An understanding of data file organization and access techniques.

8. An understanding of problem solution utilizing flowcharts, decision tables, translators, generators, programming structure, and fundamental process of analysis.

9. An in-depth understanding of a data processing system's "life cycle."

10. An understanding of data base management, data base file structure, the data base environment, and elements of a data base language.

11. The ability to program in an on-line environment.

12. An understanding of mini computers and mini computer networks.

13. A knowledge of and experience with CAI (Computer Assisted Instruction) as a learning tool.

14. Participation in an internship or work experience program.

II.2.2 Data Processing Related Occupations

As previously stated, students in this category normally have not completed their data processing training. Typical jobs for these students include: a) data control clerk; b) tape librarian; c) documentation librarian; and d) EDP salespersons. Two levels of objectives are included. Level I is considered to be the minimum training to be effective in one of the preceding occupations. Level II objectives are recommended for those students who will have a greater degree of responsibility on the job.

A. LEVEL I -- Introduction to Data Processing

1. A comprehensive understanding of current computer hardware, software, and operating environment in order to perform business functions in coordination with data processing personnel.

2. An understanding of the data processing system's "life cycle."

B. LEVEL II -- Introduction to Programming

1. A comprehensive understanding of current computer hardware, software, and operating environment in order to perform business functions in coordination with data processing personnel.

2. An understanding of the data processing system's "life cycle."

3. The ability to program effectively in at least one computer programming language.

The objectives presented in this section are suggested with respect to the currently identified needs of data processing installations in the State of Washington. They do not address all of the curriculum objectives that may be unique to an occupational program in a given geographical area. Each of the campuses offers a similar sequence of courses with instructional emphasis varying by geographical area and based upon the input of local technical advisory committees.

II.3 PROBLEM SOLVING

Problem solving is the use of computing resources to define, describe and solve those problems which can be expressed in arithmetic or logical notation. This type of problem is typically encountered in the business, natural science, social science and engineering curricula. Problem solving applications require a range of resources that span the entire spectrum of software, hardware, and peopleware. The necessary computing resources are determined by

the type of problem. With the advent of CAI, CMI, and CBI, the techniques of problem solving can be extended into the humanities and other areas traditionally untouched by computer technology. This area is designed to support disciplines other than occupational data processing, the objectives and curriculum must be tailored to fit requirements of the individual disciplines. General objectives are suggested from which may be extracted those which will satisfy specific needs. Typical disciplines that have been supported include: 1) Accounting; 2) Banking programs; 3) Education; 4) Electronics; 5) Engineering; 6) Engineering Technology (including drafting); 7) Hotel-Motel Management; 8) Management; 9) Mathematics; 10) Science; 11) Secretarial Science; 12) Social Science; 13) Statistics; 14) Allied Health Occupations, 15) Business, and a host of others.

In order to solve applications problems the student will need to understand basic computer concepts, to have some programming skills, to understand computer-assisted problem-solving, and to know how to apply those methods to his major filed.

Specific curriculum objectives are;

1. To teach a modern, high-level programming language where each student writes and processes several computer programs.

2. To use interactive time-sharing computer terminals as a training component where each student writes at least one program in a time-sharing mode.

3. To have computer facilities available for use in all instructional programs to:

   a) Provide sophisticated techniques of handling any subject matter;

   b) Expose students and faculty to new advances in data processing technology.

4. To offer a computer programming course which emphasizes problem solving techniques.

5. To provide real life situations for homework problems and student special projects via computer simulations.

6. To gain an understanding of problem solution utilizing flowcharts, decision tables, programming structure, and fundamental processes of analysis.

7. To enable the student to gain a knowledge of the capabilities of computing which will become increasingly important in gaining employment in a computer-oriented society.

8. To have video tapes, voice cassettes, and programmed instruction computer courses available for special interest or small classes.

II.4 GENERAL INTRODUCTION TO COMPUTERS AND COMPUTING

General introduction to computers and computing covers those courses based on the following themes:

1. An overall view of the computer and its applications focused on the fundamental concepts of hardware and software;

2. A study of the impact of computer technology upon society and society's problems.

3. An investigation of the extent to which the usage of computers has dehumanized the individual and ways of counteracting these effects.

4. The future recreational and potential personal growth possibilities of computer development via games, music, artificial intelligence, etc..

Specific curriculum objectives are:

1. To provide a course on the computer and society to help the community adjust to and take advantage of computer technology.

2. To provide a single course that would sufficiently acquaint students with enough aspects of data processing so that students could enter any advanced data processing course or use data processing principles in other course work.

3. To offer a survey course in automated data processing including an orientation to computer processes, hardware and software.

4. To provide real life situations for homework problems and special projects via computer simulations.

5. To foster an awareness of how the computer works, what it can do, what it cannot do, how it is misused, and how to use this technological advancement in a human environment.

6. To gain an understanding of problem solution utilizing flowcharts, decision tables, programming structure, and fundamental processes of analysis.

7. To enable the student to gain a knowledge of the capabilities of computing which will become increasingly important in gaining employment in a computer-oriented society.

8. To obtain an understanding of a data processing system's "life cycle."

II.5 COMPUTER-BASED INSTRUCTION

Computer-based instruction is the adjunctive use of computer programs (models or simulations) to assist students by providing a vehicle for conveying knowledge, by allowing practice in applying concepts, and by increasing interest through active participation in

analysis and decision making. Although computer-based instruction requires the use of a computer, no detailed knowledge of computers or computer languages is necessary, but rather a knowledge of factors influencing the model/simulation basic to the discipline itself. By using the computer to do some of the "busy work" in number computation, a student is given more time to test alternate decisions or to pursue analysis of the problem in greater depth. Also by seeing which decisions are made by which subset of the class, an instructor can evaluate his own effectiveness. It allows a student an introduction to the use of the computer in his own discipline. Current applications have been developed in a variety of disciplines, e.g., business, economics, political science, chemistry, biology, forestry, mathematics, engineering, health services.

Most normal teaching methods consist of lecture, homework problems, tests, and in some areas, laboratory experience. While computer-based instruction does not relieve an instructor or student of responsibility in the traditional areas, it does provide an alternative exercise in the use of lecture and reading knowledge to problem solving. In a survey conducted at Lower Columbia Community College, of 158 students which had used CBI in a business course, 130 or 82 percent found the use of CBI as "informative and enjoyable" while only 10 percent found CBI as "confusing or difficult." The application of theory to practice traditionally has been by either case studies or mathematical computation problems. A dificulty with the case study technique is that it isolates the general theory currently being taught by forcing too narrow a scope of application. Mathematical computations usually cannot be very sophisticated, particularly at the community college level. The computer-based simulation model avoids both of these difficulties by allowing the use of sophisticated computational procedures with a wide variety of problems. In essence, the computer does the arithmetic in a very rapid fashion, allowing the student to see results of his analyses and decisions many times during a course, rather than just once or in a very simplistic way.

Specific application objectives are:

1. To augment the normal teaching materials
2. To encourage the application of theory.
3. To expand student interest in the discipline.

II.6 COMPUTER-MANAGED INSTRUCTION

Computers may be used in the management of the learning process in two ways. First the computer is used as an aid in the monitoring and recording of student performance over learning objectives. In the second, the computer assists the instructor in testing and prescribing appropriate materials to each student.

Teachers must keep records of student performance and achievement and use the records for student and curriculum evaluation. The computer's clerical and analytical abilities are likely to be of value in these tasks, particularly when the classes are large in size.

The production and grading of tests, the analysis of their results, the evaluation of individual and group student performance, and class record keeping are among the most time-consuming and least-valued aspects of the instructional job. Research has shown that frequent testing with appropriate feedback of results enhances and increases comprehension and learning of material. However, frequent testing and immediate feed-back of results are not often carried out at present due to the overwhelming demands on time and clerical resources necessary. The computer's inherent nature makes it a viable tool for managing the teaching-learning process.

Also, the computer can be used to help evaluate student areas of difficulty, and direct the student to remedial materials in order to facilitate his learning of the subject matter.

Management uses of computers in the learning-teaching process have the following set of objectives:

1) To enhance and support individualized instructional offerings, experiences, and evaluation techniques

2) To assist the instructor in determining curriculum objectives, in evaluating student performance, in reducing time-consuming clerical tasks done manually

3) To increase the relationship between course content, testing materials, and student self-evaluation of progress through each unit of learning

4) To reduce the dependence of the student on the instructor for testing, remedial references, and retesting

5) To improve the effectiveness with which course materials are conveyed to both large and small classes students with a wide range of aptitudes and backgrounds

6) To increase the opportunity for common terminology, common course objectives, and common measurement instruments within the community college system.

II.7 COMPUTER ASSISTED INSTRUCTION

Computer Assisted Instruction involves the use of computing resources in the delivery and presentation of course content to the student.

CAI has as objectives:

1. To provide individualized instruction for each student with the flexibility to adapt to each student's unique learning characteristics.

2. To enhance teacher productivity.

3. To allow the instructor to concentrate on the human aspects of the student-teacher relationship.

4. To improve the quality of the learning process through increased motivation and deeper understanding of primary concepts.

The results of recent CAI trials and the scope and sophistication of some on-going programs including two major efforts to demonstrate the feasibility of CAI on a large-scale, suggest that a computer may indeed come to play an important role as teacher, though probably more slowly than originally anticipated.

II.8 COMPUTER SCIENCE MAJOR

Computer Science is a field of inquiry devoted to the representation, storage, manipulation and presentation of information in an environment permitting automatic information systems. The computer scientist is interested in discovering the means by which information can be transformed in order to model and analyze the information transformations in the real world.

The objective of Computer Science education is to develop professionally competent and briadly educated computer scientists. At the undergraduate level, the preparation for entering a professional career or for pursuing graduate studies is provided. Undergraduate education is not designed to prepare students for specific jobs, but rather to provide the basic knowledge for any number of present and future jobs; especially important is a foundation that will not become obsolete as technology advances and changes.

III CURRICULUM CONTENT

III.1 OVERVIEW

Curriculum content guidelines which achieve the curriculum objectives for each of the seven areas of computing activity are presented in this chapter. Clearly, individual instructor perogative, local constraints and emphasis, plus student intent will tend to diversify curriculum content over the community college system. Therefore, it is expected that the curriculum content and application characteristics specified herein are to be used primarily as guidelines and not as absolutes.

III.2 DATA PROCESSING TECHNOLOGY

The curriculum content and courses being offered by the vocational data processing programs must meet a wide variety of needs. As such a given course must provide training for not only the data processing major, but for the student in other disciplines. This section will address typical courses being offered at the various community colleges with data processing programs. The matrix that follows indicates those courses and which of the curriculae are being satisfied by the specific course. Following that, each course will be discussed with respect to course objectives and content.

COURSES	CURRICULA SATISFIED			
	Data Entry	Comp Operatn	Comp Prog	Support
1. Keypunch Theory & Application	X			
2. Introduction to Data Processing	X	X	X	X
3. Computer Operations Theory & Application		X	X	
4. Computer Programming Fundamentals		X	X	X
5. Job Control Language & Operating Systems		X	X	
6. Computer Programming / RPG		X	X	
7. Computer Programming / ANS COBOL		X	X	X
8. Computer Programming / ALC			X	
9. Systems Analysis			X	
10. Structured Programming			X	
11. Data Communications			X	
12. Computer Programming / BASIC			X	X
13. Computer Programming / FORTRAN			X	X
14. Computer Programming / PL/1			X	
15. Data Base Management			X	
16. Mini Computer ORIENTATION		X	X	X
17. CBE AUTHOR LANGUAGE				X
18. Internship / Work Experience	X	X	X	

III.2.2 Curriculum Content Guidelines

1. Key Punch Theory and Applications

Course Objectives

At the completion of the course, the student should be able to:

a) power on and ready the Data Entry machine;

b) prepare a program card;

c) determine machine malfunction and recommend correction procedures;

d) punch 10,000 strokes/hour on an IBM 029 with an error rate not exceeding 4 percent;

e) punch 12,000 strokes/hour on a buffered data entry device with an error rate not in excess of 4 percent.

III.2.2.1.3 Course Content

Introduction to key punch procedures; program card preparation; familiarization with skills necessary for quick and safe operation of equipment; written and key punch tests; building of speed and accuracy; production work.

2. Introduction to Data Processing

Course Objectives

At the completion of this course, the student will be able to:

a) understand basic data processing terminology;

b) understand data flow concepts;

c) describe various methods of handling data including manual, electromechanical and computer-oriented approaches.

d) describe software, hardware and peopleware as related to data processing;

e) discuss some of the effects of the computer in society.

Course Content

Introduction to data processing terminology, unit record processing, computer hardware, computer software, problem-solving considerations, computer systems and applications, methods of accessing the computer, computers and society.

3. Computer Operations Theory and Application

Course Objectives

At the completion of the course, the student shall be able to perform the required tasks of a computer operator at the job entry level.

Course Content

Introduction to the functions of the console, central processing unit and input/output devices. Introduction to operating systems, power procedures, console operation, peripheral operation, work schedules, personnel safety, file maintenance, time utilization and work efficiency.

4. Computer Programming Fundamentals

Course Objectives

At the completion of this course, the student will be able to:

a) Solve problems using Algorithmic Methods and flowcharting techniques.

b) List and describe the operations and the major hardware components of a computer system.

c) Convert decimal numbers, alphabet and special charters into the computer representation of data for various computer models by the polynomial expansion method; BCD and EBCDIC methods.

d) Demonstrate comprehension of the BASIC computer language by writing and running on a computer, at least five programs.

e) Explain how computers have affected and may affect our society.

f) Describe the function(s) of software in the total computer environment and discuss the more common languages utilized by computer programmers.

Course Content

Introduction to Problem-Solving Concepts, Introduction to Computer Systems (hardware and software), Computer Information Representation, Introduction to Business Problem Solving, The Computer and Society, The Basic Language.

5. Job Control Language and Operating Systems

Course Objectives

At the completion of this course the student should be able to

a) effectively use the library facilities of a given operating system;

b) use the file-to-file utilities of a given operating system;

c) code the control statements that will make efficient use of a sort program;

d) develop job streams using card, tape, and disk files.

Course Content

Introduction to operating systems, data management concepts, control statement techniques, system control and service programs, application programming considerations.

6. Computer Programming/RPG

Course Objectives

At the completion of this course, the student will be able to:

a) describe and use the various RPG coding specifications.

b) write RPG programs having:

1) the four basic arithmetic functions;

2) multiple level control breaks;

3) table handling;

4) tape input and output.

c) debug RPG programs using the diagnostics provided by the compiler.

Course Content

Introduction to RPG, RPG coding specifications, basic input/output, arithmetic statements, multiple level control totals, tables, tape handling, use of indicators.

7. ANS COBOL Language

Course Objectives

At the completion of this course, the student will be able to:

a) describe and use the syntax of COBOL:

b) perform data manipulation;

c) utilize the various file or organization techniques as they relate to card, printer, magnetic tape and direct-access files;

d) program segment - to facilitate more efficient use of core storage through the use of the segmentation feature of COBOL;

e) Utilize the Sort feature incorporating input and output procedures; and

f) debug source and object programs using the debugging language, compile output, prepare flow charts and read core dumps.

Course Content

Introduction to COBOL; COBOL fundamentals, table-handling, COBOL for tape devices, DASD organization, sort and subprogram concepts. It is recommended that this course be two quarters in length.

8. Computer Programming/Assembler Language

Course Objectives

At the completion of this course, the student will be able to:

a) describe the assembler language instruction set;
b) describe the various data structures and formats;
c) write assembler language programs using decimal and fixed-point instructions;
d) read a core dump.

Course Content

Introduction to assembler language, internal data representation and data structures; standard instruction set with emphasis on decimal and fixed-point operations, supervisor and I/O macros, reading of core dumps, program segmentation.

9. Systems Analysis

Course Objectives

At the completion of this course the student should have an understanding of the duties of the systems analyst together with an understanding of the specific methods and techniques for conducting a systems project, including the following phases;

a) preliminary investigation and feasibility study;
b) design of systems output;
c) design of systems input;
d) design of systems files;
e) design of systems processing;
f) design of systems controls;
g) systems project management;
h) programming specification;
i) programming, testing, and documentation;
j) systems implementation and evaluation;

Course Content

Role of the Systems Analyst; business as an information system; coding; forms design; charting techniques; flowcharting; communications; study phase; system performance; feasibility analysis; study phase report and review; system design.

10. Structured Programming

Course Objectives

At the completion of this course the student will be able to:

a) differentiate between modular, structured and traditional programming; techniques.

b) prepare programs using the structured programming techniques; and debug them; A comparison of "normal" programming techniques will be demonstrated by the student.

Course Content

Modular programming techniques; top-down or structured programming techniques and how they compare with traditional methods; structure of the flow of control of a program in structured programming; verification of each level of a structured program.

11. Data Communications

Course Objectives

At the completion of this course the student will be able to:

a) understand teleprocessing terminology;

b) describe various access methods to a computing facility;

c) design a simple teleprocessing system;

d) discuss equipment considerations relating to a proposed teleprocessing system; and

e) write and execute interactive programs.

Course Content

Data communications equipment, signal characteristics, channel categories, modulation, transmission modes, common carriers, error detection, network structure, and utilization.

12. Basic Language

Course Objectives

At the completion of this course the student will be able to:

a) utilize a terminal (TTY, CRT, etc.) to send and receive the basic programs;

b) demonstrate comprehension of the Basic Language by writing and running on a computer at least five application programs, including documentation.

Course Content

Introduction to computer and terminals; syntax of Basic; various Basic instructions; use of library functions.

13. ANS Fortran IV Language

Course Objectives

At the completion of this course the student will be able to:

a) describe and use the syntax of Fortran;

b) describe the practices and pitfalls of Fortran;

c) analyze, program, test, debug and document at least five Fortran programs;

d) prepare, through keypunch or edit file, these programs and compile and execute them on a computer.

Course Content

The elements of Fortran; input and output; arithmetic programming; control programming; logical programming; arrays. Subprograms; and, internal data representation. Assumes that the student has completed at least a college-level algebra course.

14. PL/I Language

Course Objectives

At the completion of this course the student should be able to:

a) describe and use the syntax of PL/I;

b) read and explain a given program written in PL/I;

c) analyze, program, compile, test, debug and document several PL/I programs;

d) design a structured program using PL/I.

Course Content

Structural elements of PL/I; input and output; data manipulation; computational and manipulative facilities of PL/I; looping; program design based on "top-down programming."

15. Data Base Management

Course Objectives

At the completion of the course the student will be able to:

a) define a data base;

b) define an outline architecture for a data base;

c) discuss the elements of a data base language;

d) discuss various physical organization methods of data base implementation;

e) update and retrieve data from a data base.

Course Content

Data management; data base environment and files; elements of a data base language; and the advantages and limitations of the "data base" approach.

16. Mini-Computers

Course Objectives

At the completion of this course the student will be able to:

a) program and interface with mini computers;

b) discuss and utilize logical operations and simple digital circuits;

c) discuss and utilize instructions commonly used in small computers, including addressing techniques;

d) load, start and execute computer programs on a mini computer.

Course Content

Number system and digital codes; logical operations and digital circuits; basic computer instructions; review of programming fundamentals; mini computers; data simulation.

17. Author Languages

Course Objectives

Upon completion of this course, the student will be able to:

a) utilize a small programmed learning text consisting of a set of English based words that direct the computer;

b) know how author languages function and know the elements of a lesson;

c) develop a course using an author language.

Course Content

Various author languages; what is an author language; developing the elements of a lesson plan; planning on a computer; debugging and updating the course.

18. Internship/Work Experience

Course Objectives

At the completion of this course the student will be able to:

a) develop and implement a specific data processing project;

b) to apply current programming techniques with a minimum of supervision and instruction;

c) to document the completed project.

Course Content

Provides advanced data processing students with actual work experience parallel to the actual job environment. Individual programs of work and classroom instructions are fitted to the student's needs and desires.

NOTE:

Not all of the courses described in the preceding pages are offered at all of the fifteen community colleges offering data processing technology. Data entry training and computer operations training, for example, are not approved programs at many of the fifteen colleges. Author language training can only be offered at colleges with access to a computing facility having the necessary software. Based upon input from the curricula at the fifteen colleges having data processing training and input from local technical advisory committees, the courses included in this section are considered to be current state-of-the-art.

III.3 PROBLEM SOLVING COURSES

The curriculum content of problem solving courses is based upon the utilization of a modern, high-level programming language. This necessitates acquiring the concept of problem identification, the formulation of an algorithm, and mastery of the syntax and logic of whatever programming language is being used. In addition to specific courses in problem solving and programming, educators are now finding that the computer can be utilized in virtually any academic discipline as a problem solving tool.

Problem-solving and the academic use of computers can be dichotomized into computer-programming courses and courses which use computer programs.

In the first category, typically, we have Fundamentals of Computer Programming, BASIC programming, FORTRAN programming, & COBOL programming.

1. Fundamentals of Computer Programming

Course Objectives

At the completion of this course the student will have a working knowledge of the following programming concepts:

a) List and describe the operations and basic hardware components of a computer system (input devices, compilers, CPU, Output devices and other peripheral equipment);

b) The utilization of a terminal (TTY, CRT, etc.) for interactive computing and methods of batch mode computing;

c) Problem solving techniques (problem identification, problem analysis, algorithmic processes, flow charting, program structure, coding, program execution and program debugging;

d) Machine languages and programming languages including the ability to write a simple program in at least one modern programming language (BASIC, FORTRAN, COBOL, etc.);

e) The ability to apply those concepts to the solution of simple business-type problems.

Course Content

Elementary programming concepts including problem analysis and an introduction to a modern programming language. A brief introduction to automated data processing covering the basics of computer hardware and software.

2. BASIC Programming

Course Objectives

At the completion of this course the student will have a working knowledge of the following programming concepts as applied through the BASIC language:

a) the utilization of a terminal (TTY, CRT, etc,) to prepare and submit a computer program;

b) the basic components of a computer system (input devices, compilers, CPU, output devices, and other peripheral equipment);

c) the source program and the object program;

d) errors (syntax, logical and blunder);

e) BASIC statements (REMARK, assignment, conditional transfer, unconditional transfer, input, output, STOP and END); and

f) programming concepts (line numbers, constants, variables, expressions, arithmetic operators, relational operators, loops, subscripted variables, functions and subroutines);

g) BASIC commands (LIST, SCRATCH and RUN); and

h) solution of practical problems by writing computer programs (flow charting, writing, coding and debugging).

Course Content

An introduction to computer programming emphasizing the use of the computer as a tool for problem solving. Programs are written in the BASIC language and processed in either an interactive or batch mode.

3. FORTRAN Programming

Course Objectives

At the completion of this course the student will have a working knowledge of the following programming concepts as applied in the FORTRAN IV language:

a) the preparation and submission of a computer program via an interactive mode (TTY, CRT, etc.) or batch mode;

b) the basic components of a computer system (input devices, compilers, CPU, output devices, and other peripheral equipment);

c) the source program and the object program;

d) errors (syntax, logical and blunder);

e) FORTRAN statements (COMMENT, ASSIGNMENT, CONTINUATION, DIMENSION, unconditional transfer, arithmetic IF, logical IF, computed GO TO, DO, FORMAT, input, output, STOP, RETURN, and END);

f) programming concepts (labels, constants (real, integer, string), data variables (real, integer, string), expressions, operators (arithmetic, relational, logical), loops, arrays, functions and subroutines);

g) analyze flow chart, write, code and debug a minimum of five FORTRAN programs;

h) prepare through key punch or edit file, these programs and compile and execute on a computer.

Course Content

An introduction to the FORTRAN programming language and its application to the solution of problems in the areas of business, mathematics and other natural and social sciences.

4. COBOL Programming

Course Objectives

At the completion of this course the student will have a working knowledge of the following programming concepts as applied in the ANSI COBOL language:

a) the preparation and submission of a computer program in the batch mode;

b) the basic components of a computer system (input devices, compiters, CPU, output devices and other peripheral equipment);

c) the source program and the object program;

d) errors (syntax, logical and blunder);

e) COBOL statements (simple, compound, imperative, conditional, input (OPEN, READ, ACCEPT), output (WRITE, CLOSE, DISPLAY), arithmetic (ADD, SUBTRACT, MULTIPLY, DIVIDE, COMPUTE), data transmission (MOVE, EXAMINE), sequence control (GO TO, PERFORM, ALTER, STOP), RUN and END);

f) programming concepts (characters, names (data procedure, condition, special), works, constants (literal, figurative), operators (arithmetic expression, relational expression, logical expression), punctuation symbols, sheet format, sequence numbers, continuation, divisions (identification, environment, data, procedure), sections, paragraphs and entries);

g) utilize various file organization techniques as they relate to card, printer, magnetic tape and direct-access files;

h) debug source and object programs using the debugging language, compile output, prepare flow-charts and read core dumps;

i) An introduction to the ANSI COBOL programming language and its application to the solution of typical business problems. Emphasis on magnetic disk file organization and magnetic tape processing.

Course Content

An introduction to the ANSI COBOL programming language and its application to the solution of typical business problems. Emphasis on magnetic disk file organization and magnetic tape processing.

In the second category of problem solving with a computer, we have academic courses which utilize computer programming to solve problems relative to a specific descipline.

Following is a selected list of subject areas and several computer applications for each area. The primary uses of the computer in classes other than mathematics are for problem solving, simulation, and drill/tutorial. All of the computer programs suggested are already in various existing libraries. Few examples are given in mathematics because they are so numerous and most mathematics instructors are aware of the computer capabilities within their discipline. See Reference (CCUC vols 1-6)

1. art - computer animated films; computer aided graphics as an art form; the microfilm plotter and computer art;

2. biology - heredity; photosynthesis; pest ccntrol; genetic modules; enzyme action; computing in animal physiology; testing program in audio-tutorial biology; prey-predator simulation and analysis; data analysis; pollution models;

3. business - inventory; depreciation; tax models; budget forecasting; bank statements; invoicing; mortgages; interest; economic models; population and economic growth; mathematics of finance; simulation of financial statements;

4. calculus - Newton's method; curve fitting; numerical integration; numerical differentiation; approximation of functions;

5. chemistry - data analysis; chemical equilibrium; testing of chemical names and formulas; aromatic organic synthesis; conversions; titration; enzyme reactions; CAI and CBI in organic chemistry;

6. education - CAI in early identification of handicapped children;

7. English - simulation of poetry explication; synonym drill; computer teaching of literary stylistics; "From poetry to politics"; CAI in linguistic theory;

8. foreign language - computer generated video tape for teaching French grammar; drilling Spanish verb forms; sentence generation through visual cues; computer-supplemented Latin instruction; conjugation drill; Spanish vocabulary drill;

9. game theory - dynamics of environmental systems; urban planning; qualitative organic identification; economic policies; comparative shopping; city government;

10. music - CAI in music theory; synthesizers and the physics of music; digital system for the instruction, composition and performance of music; an interactive computer music system;

11. physics - computer graphics; geometrical optics; wave motion; simulated projectile motion; simulated thermodynamics;

12. psychology - simulation of psychotherapy results; computer applications in experimental psychology; training in experimental inquiry;

13. social sciences - population dynamics; election models and prediction; political simulation models; analysis of survey data; CAI in American history; teaching spatial organization;

14. statistics - Monte Carlo experiments; hypothesis testing; regression analysis; correlation; descriptive statistics; analysis of variance;

III.4 GENERAL INTRODUCTION TO COMPUTERS AND COMPUTING

Any student, no matter what discipline, may benefit from a knowledge of computers, computing, and electronic data processing. An understanding of the concepts and applications of computers and automation to the problems and processes of modern civilization should be a part of general education. Some knowledge of computers and programming concepts will be necessary to function in the modern business world and to function as a citizen and consumer in the society of the future. It is essential that an individual understand the capabilities and limitations of computers as well as their impact on economics, technology and individuality.

1. Introduction to Data Processing

Course Objectives

At the completion of this course the student will have a knowledge of the following data processing concepts:

a) basic data processing terminology;

b) the history and evolution of computers;

c) decision-making processes;

d) systems analysis and design;

e) program flow charting;

f) punched card data processing (reproducer, interpreter, sorter, collator, calculator and tabulator);

g) input/output devices;

h) computer hardware and software;

i) data entry systems;

j) various methods of handling data including manual, electromechanical and computer-oriented approaches;

k) the data processing cycle.

Course Content

Basic methods, techniques, and systems of manual, mechanical and electronic data processing principles. Fundamental methods and techniques of punched card, and magnetic disk systems. Unit record processing. Computer hardware and software.

2. Computers and Society

Course Objectives

At the completion of this course the student will have a knowledge of the following computer concepts:

a) the history of the computer and an understanding of computer systems;

b) representation of information and processes;

c) language and information retrieval;

d) models and simulation;

e) computers and education;

f) games, logic and the computer;

g) thought processes and artificial intelligence;

h) automation and the economics of computers;

i) computers and social responsibility;

j) the computer and an individual's right to privacy.

Course Content

A general course investigating the capabilities, limitations and problems associated with computers. An important objective is to acquaint the student with the vocabulary, major concepts and the social implications of the computer.

Introduction to Computers, Computing and Computer Science

Course Objectives

At the completion of this course the student will have a working knowledge of the following concepts:

a) the operations and basic hardware components of a computer system (input devices, compilers, CPU, output devices and other peripheral equipment);

b) the utilization of a terminal (TTY, CRT, etc.) for interactive computing and methods of batch mode computing;

c) problem solving techniques (problem identification, problem analysis, algorithmic processes, flow charting, program structure, coding, program execution and program debugging);

d) machine languages and programming languages including the ability to write a simple program in one modern programming language (BASIC, FORTRAN, COBOL);

e) the ability to apply these concepts to the solution of a simple problem;

f) the history and evolution of computers.

Course Content

Emphasis on the basic computing system, organization of the most important computers, hardware and software features of the most popular computers and comparison of computer systems. Elementary programming concepts including problem analysis and an introduction to a modern programming language.

III.5 COMPUTER BASED INSTRUCTION - APPLICATION CHARACTERISTICS

Computer based instruction is generally defined as the use of specialized computer programs such as simulators, emulators, and models as tools to support the instruction of traditional disciplines such as economics, business, engineering, forestry, and health education. As a simulator of the real world the computer is used as a vehicle for processing a mathematical model of the environment which is the subject matter of the course. The course content is primarily based upon the simulated environment created in the mathematical model. According to Naylor, et al (1966):

> "Simulation is a numerical technique for conducting experiments on a digital computer, which involves certain types of mathematical and logical models that describe the behavior of a business or economic system (or some component thereof) over extended periods of time." (Naylor, T.H., J.L. Bolintfy, D.S. Burdick, and K.Chy, Computer Simulation Techniques, New York: John Wiley & Sons. 1966)

In short, computer based instructional applications consist of computer models that are important adjuncts to the instructional process in at least three ways. First, they provide a means for conveying to the students a body of substantive knowledge. Second they enable the student to practice applying his theoretical knowledge in a realistic environment, closely resembling real-life situations, and third they heighten the student's interest in the subject matter.

III.6 COMPUTER MANAGED INSTRUCTION APPLICATION CHARACTERISTICS

Computer managed instructional applications participate in the instructional process by effecting and organizing each step of the process. CMI provides information on a basis of which instructional-sequence decisions can be made. There are four characteristics of any computer-managed instruction system:

1) a file of student characteristics and performance;

2) a set of diagnostic examinations;

3) a set of instructional segments or modules;

4) a computing resource that uses data on student characteristics and performance through diagnostic examinations to prescribe an appropriate instructional module.

Computer-managed instruction differs from computer-assisted instruction in that CMI is adjunctive to the learning process, not the vehicle of presentation of course content as is CAI. CMI differs from CBI in that CMI manages and directs the sequence of learning events, and then records the progress of each student through the sequence; CBI does not include the management or the recording of progress.

III.7 COMPUTER ASSISTED INSTRUCTION - APPLICATION CHARACTERISTICS

The term CAI is reserved for applications in which the computer program controls the amount and sequencing of information given the student through direct student/program interaction. The CAI computer program itself is designed to drill or teach the student. The CAI program provides drill through a sophisticated form of programmed instruction (PI). A frame of information followed by a question is presented to the student either on the typewriter or on the display device (CRT or plasma display tube). The student responds via the typewriter, or with light pen, and based on his response data, he receives feedback concerning his answer followed by further information.

The potential for flexibility in a CAI program is almost unlimited. Unfortunately, the reality falls far short of the potential. Suppes (1969) identifies three levels of CAI according to the degree and complexity of student-program interaction. These are a) drill and practice, b) tutorial, and c) dialogue.

Drill and Practice:

When the computer is used to provide drill in a skill such as addition or spelling, the lowest level of student-program interaction is involved. A problem is given the student; he responds; he is told whether or not he is correct; and he is given a new problem. Most existing CAI programs are at this level.

Tutorial:

A CAI program designed to teach a concept, value or skill is classified as tutorial. CAI tutorial programs borrow heavily from the theory of Programmed Instruction. CAI does have many (potential) advantages over PI in speed of branching, gathering of data for branching, collection of student performance data and ease of revision.

Dialogue:

The dialogue level of CAI "exists primarily at the conceptual rather than the operational level," according to Dr. Suppes (1969). Emphasis will be on cognitive learning at complex levels, such as heuristics, analysis, and synthesis. Provisions can also be made for voice and/or Braille communication between the student and the computer. This will help the handicapped student.

Students need drill and repetition to learn most concepts and skills. The amount needed varies among students on a given concept. A good CAI program must provide enough drill for any student, but should not require drill beyond a student's needs. A fairly sophisticated program, developed and refined based on large amounts of formative data, is needed to do this effectively.

Implementing CAI at the Drill and Practice level will most likely occur before the other two levels since it is the least costly to develop. A fairly substantial body of Programmed Instruction research has been built up in the last 15 years. Most of the results are directly applicable to CAI; results such as the value of low error rate, immediate feedback to the student, and active involvement of students in the instructional process. Furthermore, techniques for program development, revision, and evaluation in PI can be used in CAI.

The implementation of CAI in mainline instruction, i.e., replacing some or all of the usual teaching staff, has proved to be a task of major magnitude. Costs of program development per hour of tutorial instruction run as high as $20,000. At best, using resources currently available, good tutorial CAI programs can be produced for somewhere in the neighborhood of $3,000 per hour, according to C. Victor Bunderson (1970). This assumes the existence of languages, systems, and authoring techniques that are still in the development stages.

The justification for the time and expense required to develop high quality CAI programs comes from the potential for greatly increased effectiveness and the economics of mass distribution. CAI evaluation studies have shown that students usually do at least as well under CAI as under traditional methods. One should not expect to be able to measure striking improvements in the performance of the better college students by means of CAI. "A" students will work until they get "A's." "B" students "B's", and other students until they reach their level of aspiration. It is difficult to measure the time and effort they spend outside of class or outside of CAI to accomplish this. The most striking finding in CAI evaluation studies has been that students are able to achieve educational objectives in much less time. Savings of 40 percent or more are not uncommon (Ford & Slough, 1970; Homeyer, 1970; Hollen, Bunderson, and Dunham, 1968). This increased efficiency has important economic implications for education. Another important finding is that lower

ability students are able to achieve important performance gains by means of CAI, often approaching the same levels as the higher ability students.

Students are able to progress rapidly in learning those skills in which they are diagnosed as being deficient. Students at all levels of ability, except the highest, showed gain from pre-test to post-test, with the more deficient students showing the most gain. Attitude toward the experience varied widely, with most students being favorably disposed. We have some evidence that the more negative attitudes were found among the poorer students, whose adjustment to any kind of concentrated study has not been good.

Few attempts have been made to implement CAI at the dialogue level. Current hardware and software capabilities appear to be sufficient to provide support at this level, however; the extra time and costs due to the more complex programming requirements have been strong deterrents to advancement in this area.

III.8 COMPUTER SCIENCE MAJOR

Computer science majors primarily study the means by which information can be transformed in order to model and analyze the information transformations in the real world.

This interest leads to inquiry into the theory of (a) effective ways to represent information of all forms, (b) effective algorithms to transform information, (c) effective languages with which to express algorithms, (d) effective hardware and software processors to execute algorithms, (e) effective means to monitor the process and display the transformed information, and (f) economic ways to accomplish the above.

The study of computing system architecture is clearly in the realm of the computer science specialist; the study of algorithms is also an area with great breadth and general educational impact. This is expressed eloquently by D. E. Knuth - as follows:

> "A person well-trained in computer science knows how to deal with algorithms: how to construct them, manipulate them, understand them, analyze them. This knowledge prepares him for much more than writing good computer programs; it is a general-purpose mental tool which will be a definite aid to his understanding of other subjects, whether they be chemistry, linguistics, or music, etc.. The reason for this may be understood in the following way: It has often been said that a person does not really understand something until he teaches it to someone else. Actually a person does not really understand something until he can teach it to a computer, i.e., express it as an algorithm. 'The automatic computer really forces that precision of thinking which is alleged to be a product of any study of mathematics.' The attempt to formalize things as

algorithms leads to a much deeper understanding than if we simply try to comprehend things in the traditional way."

Although computer science courses are not now well defined in the Community College system, some Data Processing technology courses are being utilized for the purpose of computer science transfer programs. Computer science, a developing discipline, may soon become a viable program in the Community Colleges.

IV SOFTWARE SUPPORT REQUIRED

IV.1. OVERVIEW

Computer software required for instructional computing in the Community Colleges consists of computer programs, operating systems, and data sets necessary to support the seven categories of application. Although each curiculum or application, from Data Processing Technology to Computer - assisted instruction, is relatively distinct, it is not the case that software resources required to support these areas need be distinct. In fact, it may be the case that the same FORTRAN compiler, for example, may support, in part, all of the areas.

Clearly, the data processing technology curriculum has need for a wider variety of software than any of the other areas. It has been observed that any software set capable of supporting a well-founded D.P. Technology curriculum will support most of the other areas, too.

This chapter defines the necessary software for minumum support to each of the areas of instructional computing in the community colleges.

IV.2. SPECIFIC SOFTWARE REQUIRED - DATA PROCESSING TECHNOLOGY

Keypunch Theory and Application

Specific compilers and features of given operating systems are not important in this particular area. But it is necessary to have proprietary software available in order to perform an analysis on the number of key strokes/hour. This type of software is normally written by the specific college offering a data entry program and is run on the local computing facility.

Introduction to Data Processing

Again, this course does not require specific software. However, demonstration software packages, games and simulation models are desirable for the purpose of introducing data processing concepts to the novice.

Computer Operations Theory and Application

Students enrolled in this program of study require "hands-on" experience as part of their training. As such, specific pieces of software are not as important as general software which includes: 1) an operating system; 2) production runs and procedures; 3) sort and file-to-file utilities; and 4) drill and practice software to aid the student in console operations and responses.

Computer Programming Fundamentals

Software required for this type of course primarily depends upon which computer language has been chosen as the vehicle to introduce programming fundamentals. The language most commonly used in the fifteen schools for introductory programs is the BASIC language. Therefore an interactive BASIC compiler or interpreter would be required. Those schools using a language other than BASIC, however, would have a need for a different compiler.

Job Control Language and Operating Systems

The software required for this course is primarily related to that provided by a comprehensive operating system. As part of an operating system (or externally supplied by means of software packages), software required includes: 1) system library facilities; 2) file-to-file utilities; and 3) sort/merge. Examples of appropriate operating systems include: DOS (disk operating system); OS (operating system); or combinations such as DOS/VS or OS/VS (virtual storage).

Computer Programming/RPG

Although a standard RPG compiler is probably adequate for this course, an RPG II compiler is highly desirable. This is due to an abundance of requests from both students and industry to provide training in RPG II.

Computer Programming/ANS COBOL

Required software for this course is an ANS COBOL compiler. The compiler should include special COBOL features such as the SORT verb, Report Writer and debugging facilities.

Computer Programming/Assembler Language

Any assembler language compiler would be considered adequate for this course. The purpose of this course is to provide instruction in a low-level symbolic language. Traditionally, however, the BAL language and associated compiler has been used.

Systems Analysis

Specific software is not required for this course, but sufficient software to design and implement comprehensive business applications systems is necessary. ANS COBOL is desirable.

Structured Programming

Compilers that can be utilized for "top-down" programming efforts include: a) ANS COBOL; b) FORTRAN; c) PL/I; and d)

Assembler Language. One of these compilers is a "must" for a course in structured programming.

Data Communications

Required for this course are both a compiler supporting conversational programming and an interactive text editor. Examples are interactive BASIC and WYLBUR (as used at WSU Service Center).

Computer Programming/BASIC

An interactive BASIC compiler or interpreter is necessary.

Computer Programming/FORTRAN

A full FORTRAN IV or ANS FORTRAN compiler is required.

Computer Programming/PL/I

A PL/I compiler is a minimum requirement, but PL/C is highly desirable.

Data Base Management

Although this course may be approached from the theory standpoint, it is recommended that a Data Base Management System be available for use. Such DMBS packages recommended include: a) TOTAL; b) ADABAS; c) IMS, or d) MARK IV.

Mini-Computer Orientation

Depending upon which mini-computer is the subject for study, all of the software which is a part of that mini-computer system would be necessary. Specifically, a compiler such as BASIC or FORTRAN is desirable as well as the operating system components for a selected mini-computer system.

Author Languages

Any author language package that requires a minimum amount of programming effort for the user is adequate. Specific examples include: a) TICCIT; b) PLATO; c) PILOT; d) ABLE; or e) COURSE WRITER III. A language of this type should be available at the computing resource.

Internship/Work Experience

This course, like systems analysis, does not require specific software. However, for a complete project to be designed and implemented under the internship program, a computing resource with a sizable program library is necessary. ANS COBOL is also desirable.

Specific software is required for the majority of the data processing technology courses. For the most part, a computing facility oriented toward a production environment would possess most of the software required. Access to these resources must be made available for those colleges offering data processing training.

IV.3. SOFTWARE SUPPORT FOR ACADEMIC CURRICULUM CONTENT

Software support is relatively the same for problem solving and general introductory courses as it is for CAI, CMI, and CBI applications. The Computer Science major requires a software set similar to that of the DP Technology program. It should be recognized, however, that general introductory courses may require little or no software depending upon the instructional method, and that CBI, CAI, and CMI, may require large computer programs installed in a large computing facility or small proprietory programs installed on individual classroom hardware teaching/learning stations.

Problem solving courses require the availability of compilers and/or interpreters for the FORTRAN, COBOL, and BASIC computer languages. The courses may also require the use of simulations and scientific and statistical subroutines. It is expected that most large computer centers will have a sizable library of simulation packages in the areas of marketing, finance, accounting, business policies, economics, sociology, and physical science. Also, it is probable that many instructors will want to run specific vendor

supplied programs to which they have access. These programs are almost always supplied in source code form in either FORTRAN, COBOL, BASIC, or PL/I program languages--another reason to make available these processors.

Since most student jobs for problem solving are in the "load and go" mode, it is necessary to have in-core compilers for each of the languages for academic processing.

Computer - based simulation models are usually written in FORTRAN when processed in a batch mode and are written in BASIC if implemented interactively. Necessary software for CBI includes, then, FORTRAN and/or BASIC compilers plus any specific simulation models such as MARKETING STRATEGY by Louis Boone, INTEGRATED SIMULATION by Nye, Estey, and Vines. CBI "canned" programs (pre written and pre-tested) are usually available as an instructional aid purchased with text materials, although some programs must be purchased or leased separately.

Computer software necessary to support CMI applications ranges from already-developed "canned" programs and systems to batch and interactive compilers such as FORTRAN and BASIC for development of CMI packages. For the community colleges, access to state host centers via slow-speed conversational devices and high-speed batch devices allows adequate software availability for use of most CMI programs. Some CMI systems in operation at non-host sites may require modifications to be operational at host, regional, or campus centers. Also, some self-contained, programmable terminals with pseudo-software facilities have CMI capabilities.

CAI software is divided into three parts: 1) operating systems; 2) application software; and 3) authoring software. The operating system software is concerned with task scheduling, disk management, and message handling. A general purpose operating system should be efficient enough to handle the large number of terminals necessary to deliver instruction in a cost effective manner. The applications software runs under the operating system and provides the specific CAI functions. Author software is a specific collection of special languages used for writing lessons. Currently operational example of CAI software are:

(1) PLATO - University of Illinois

(2) TICCIT - Metre Corporation & Brigham Young Univ.

(3) PILOT - University of Calif. at San Francisco

These software systems provide drill and practice, and, tutorial CAI experiences for students at institutions across the nation.

V. HARDWARE SUPPORT REQUIRED

V.1. OVERVIEW

Although hardware (the computer terminals, plotters, graphics display units, card readers, printers, punches, etc.) has often been a subject of over-concern, it is nevertheless essential that appropriate hardware be available on a campus in sufficient numbers to support the software systems being employed in the curriculum. It is not difficult to identify a minimum complement of hardware devices necessary to support a given number of students. Clearly, if a college engages in instructional computing activities that require printed reports, a device capable of printing reports is necessary. If a college engages in activities requiring conversational, interactive experience for students, then one or more interactive conversational terminals are required. The same logic can be applied to graphics and plotting activities - graphics display and plotter units will be necessary.

The following sections identify those hardware resources necessary to support the software utilized in community college instructional computing.

V.2. HARDWARE SUPPORT REQUIRED - DATA PROCESSING TECHNOLOGY

Although there is a multiplicity of computer manufacturers and State resources from which to choose, it is not within the scope of this document to recommend a particular model or vendor. Each college offering training in data processing technology must give careful consideration to total instructional use of the computing facility and its particular requirements to properly evaluate the most suitable unit, or to determine the suitability of utilizing some other computer on a time-shared basis.

However, one consideration should be foremost. If a community college is currently offering a technical program for business applications programmers and computer operators, it must make available the computer generation predominantly in use in the data processing industry. To attempt to train programmers for today's market with yesterday's equipment is unsatisfactory, and cannot produce the training level required. A commitment to train must include a commitment to proper computing resources. Not only is access to a modern computer facility imperative, but so, too, is exposure to an operating system environment. These requirements must be met in order to offer viable programs consistent with state-of-the-art technology.

A computer system that is capable of providing minimal learning experiences must be able to provide a number of systems, languages, and file processing capabilities. These would include software requirements identified in Chapter IV.

Hardware considerations will be based primarily on the institution's commitment toward integrating the computer throughout the institution. The two-year data processing curriculum will not require that sophisticated equipment be located on site, but rather that the students have access to this equipment. For a training program oriented toward computer operations, "hands-on" training is considered a must. Therefore a computing facility must be close enough for the student to have physical access. Data Entry students, too, must have access to modern data entry equipment, as specified by local advisory committees.

Specific requirements of the computing resource are:

a) Sufficient internal core storage to utilize those compilers and packages defined in Chapter IV.

b) The computing facility must have magnetic tape and disk peripherals to provide the student with file processing, sort and on-line storage experience.

c) To be able to provide adequate turnaround time so as not to disrupt the educational process. An acceptable level would range from immediate turnaround to no more than 12 hours, with a desirable average approximately 3-4 hours.

The means of access to the computing facility are varied. They include:

a) On-site computing facilities

b) High-speed RJE equipment

c) Low-speed terminals (CRT's and TTY's)

d) Any combination of the above

Exactly what hardware capability is needed by the various community colleges with occupational programs depends upon the particular circumstances of the school. Many of the existing programs have a need to upgrade their hardware, while others do not. Therefore, it will be incumbent upon each college that requires new or additional equipment to follow the resource aquisition procedures outlined in Chapter VIII.

V.3 HARDWARE SUPPORT REQUIRED - ACADEMIC COMPUTING

Hardware support for academic computing activities ranges from almost none to the remote-job-entry mini-computer supporting low speed terminals. Required access is to a spectrum of on-campus and host hardware that spans the entire width of instructional computing. For the general introductory and problem solving

courses, a slow speed terminal and/or a remote batch entry card-input, line-print output service may be necessary on campus. Host facilities should be large enough to allow storage of programs and data files in CBI, CMI, and CAI. Slow speed terminals, plotters, and graphics display terminal may be connected to remote host facilities with appropriate hardware. Optionally, transmission of data and information can be performed by bus, train, or plane. Generally, the Computer-Science curriculum requires the same facilities available to the data processing technology curriculum.

The major concern of problem solving activities is having an adequate number of teaching stations (terminals) to allow students adequate access to the computer with reasonable turnaround time.

Turnaround requirements vary course by course. Because of the student learning curve, however, the quicker the turn-around, the greater the utilization of the facilities will be.

Primarily, since most languages are supported in an interactive, conversational environment, most problem solving applications will be via teletype or similar terminals to a remote host.

Also possible, due to the developing programmable termininal and mini-computer technology, is the self-contained, interactive programmable device that supports BASIC and other languages or applications. It is possible that a combination of remote, interactive terminals and self-contained programmable terminals/minis will satisfy the problem-solving activities of a campus.

Colleges should also have access via batch facilities to large volume I/O processing for certain problem solving applications and CBI simulations that require card input and line printer output of significant volume.

Necessary hardware to support CMI applications in the community colleges range from a slow-speed terminal connected to a state host to a regional center serving two or more colleges -- that is, hardware ranges from level 1, classroom stations, through level 3, regional centers, depending upon the level of sophistication and use. (See Section V4 for description) In some cases, all that is necessary is a terminal to a host center or regional center; in others, a self-contained, programmable terminal or a campus center capable of local interactive processing will be required.

The hardware required to support CAI can vary from self-contained, programmable terminals to stand-alone on-site mini-computers to remote host systems accessed via interactive terminals. However, for other than drill and practice CAI applications, hardware support of CAI is likely to be expensive.

Hardware requirements for the Computer Science Major include access to computing power capable of supporting the necessary software. This may be achieved by an in-house system or by teleprocessing. The additional support of a mini-computer is desirable and useful but not necessary. In many cases the combination of a mini-computer as a control device in teleprocessing has proven an effective tool for study.

V.4 STRATIFIED DELIVERY SYSTEM

Colleges wishing to utilize computing facilities for instruction find the following alternatives available, they can establish their own local computer facility or participate in a remote computing cooperative, or utilize a combination of both local and remote facilities.

Advantages of establishing a local computing facility include control of software and hardware facilities, hours of operation, fixed cost for unlimited use, self-determination in direction and philosophy of computing on the campus. The advantage of fixed cost for unlimited use becomes significant where interactive computing exercises are being conducted on a large scale. Disadvantages of small computing systems include inability to support applications and environments that require a large, sophisticated software system, high costs of acquisition and maintenance for other than very small systems, and personnel costs incurred in developing adequate staffing.

The second alternative provides educational computing services to students through terminals connected to a large host computer via telephone communications network. This approach has several advantages. First, the amount of computing power to be supplied to a given school can be easily tailored to the amount of funds that are available. Second, the institution does not have to assume the task of administering and developing a large-scale computing system. Third, every student has the advantage of being able to call on the most sophisticated software systems, a feature that most institutions cannot supply on their own with small systems.

Actual experience has shown that a single terminal can expose computing to hundreds of students during the course of an academic year. However, deeper involvement in computing with more frequent exercises will require additional terminal facilities. The primary disadvantage of employing remote terminals connected to a large scale host facility is the high hourly computer and line costs incurred for relatively simple applications. For many applications involving the host accessed via terminals, the sophistication required is less than what is available. Hence costs may be higher than necessary for these simplistic applications. A second disadvantage is that the amount of computing power supplied to a given college is limited by that college's ability to pay. A third disadvantage has to do with the control of the computing resource;

changes may be forced on the remote user that are either unwelcome or disruptive.

The third alternative, a combination of both local and remote computing facilities, offers a flexible, distributed computing approach which can satisfy a majority of the computing needs of a community college at minimal cost. This alternative provides an economical approach for the initial development of an instructional program on campus through use of remote terminals in that a small initial investment for equipment is required and usage can be managed within available campus funds. Then, as computing utilization builds, it becomes possible to acquire on-campus facilities that operate in an off-line mode such that total computing costs are reduced when compared to strictly on-line mode.

A campus with computing facilities of the third alternative type can perform the majority of high-volume, introductory-type processing locally, employing the remote host facility when processing requirements exceed the local capabilities.

It is appropriate for the community colleges of this state to have computing support delivered by means of a stratified delivery structure that can be visualized as containing three levels. Each institution may have one or more configurations as depicted in the figures on the following pages.

Conceived to be an evolutionary process, the implementation of computing support on a campus shall begin with a level facility further growth to levels 2 and 3 shall be made on the basis of an orderly process that demonstrates proven ability to effectively and economically employ level 1 and 2 facilities, respectively.

V.5 DETAILED DESCRIPTION OF EACH LEVEL OF A STRATIFIED DELIVERY SYSTEM

Current State Service Centers and Eastern Washington State Community College have sufficient resources to provide the Washington State Community College system with hardware and software support for interactive and remote batch processing. Many community colleges are currently utilizing these resources to meet their increasing computing resource needs. However, as volume of processing increases at State Service Centers, other more cost effective alternatives may become available to community colleges such as stand-alone programmiable terminals and mini computers. The following Stratified Delivery System describes a coordinated evolutionary growth pattern for individual colleges to consider.

STRATIFIED DELIVERY SYSTEM

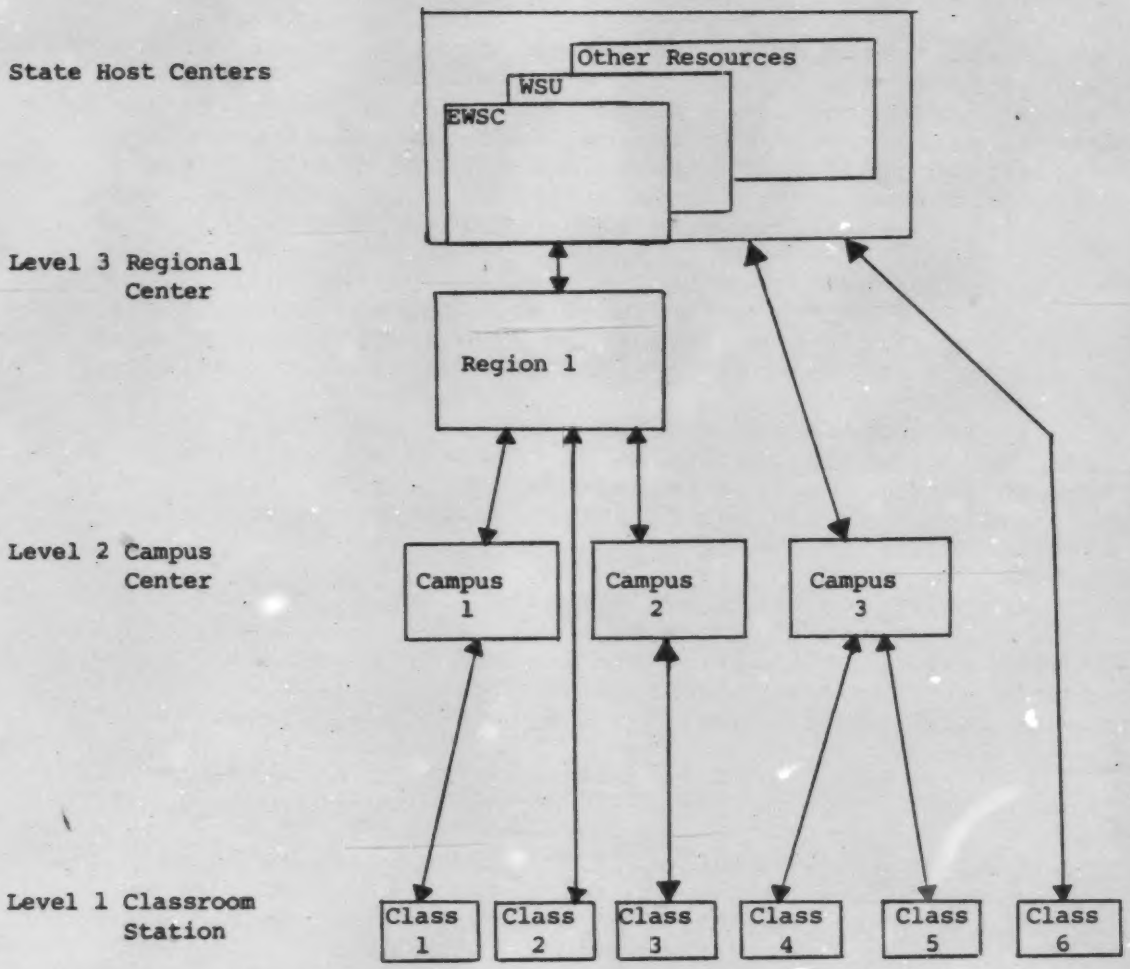
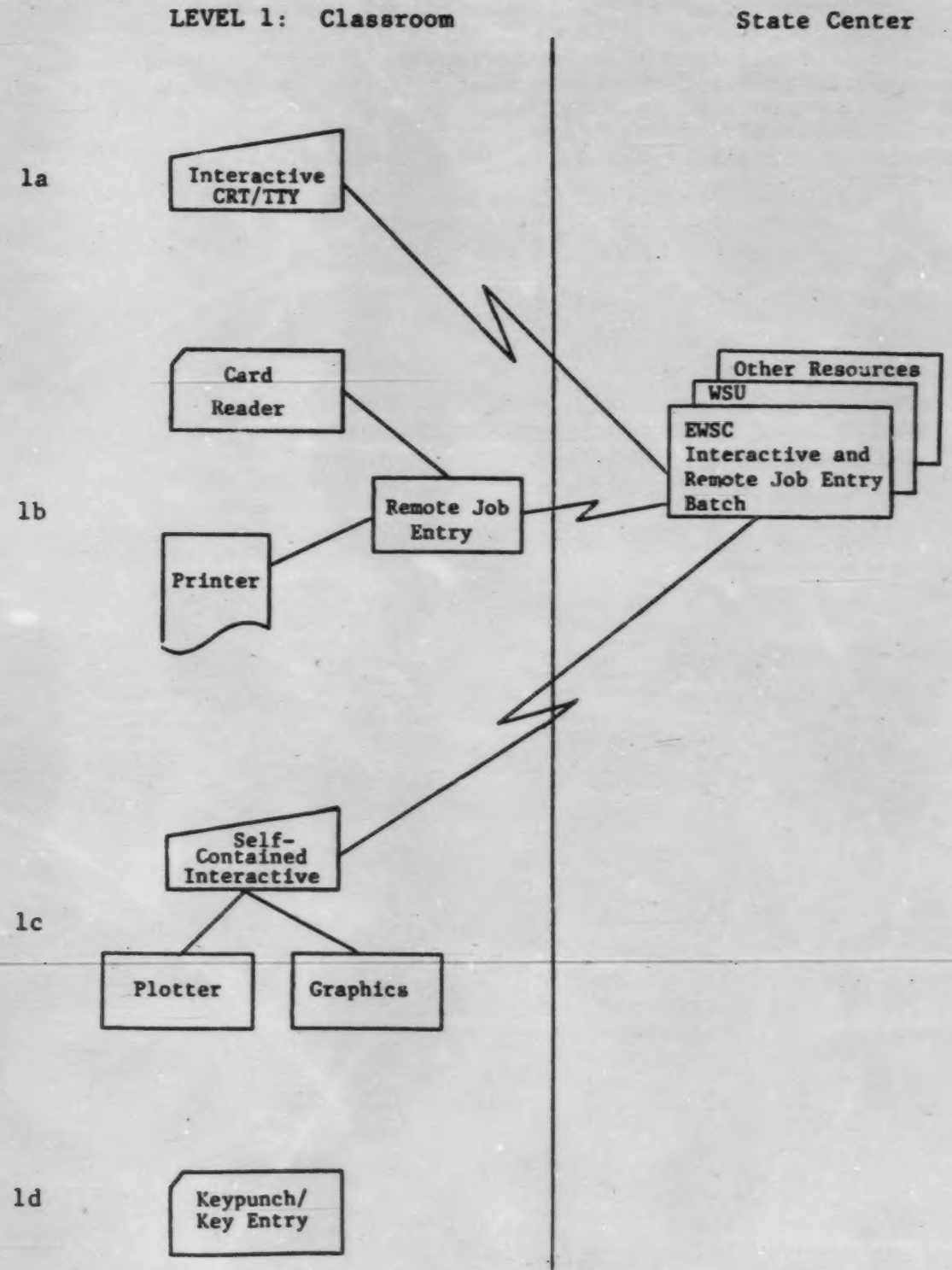


Figure 1

Level 1

At the lowest level of the stratified delivery structure is the classroom station, Level 1. Conceptually, Level 1 consists of interactive keyboard terminals connected by means of voice-grade telephone lines to a remote host facility batched remote job entry terminals, self-contained programmable interactive termiansl data entry or a combination of all four. (See figure 2.) All colleges will initially acquire, relative to their computing needs, Level 1 configurations.

Figure 2


LEVEL 1: Classroom
State Center
1a
Interactive CRT/TTY
Card Reader
1b
Remote Job Entry
Printer
Other Resources
WSU
EWSC Interactive and Remote Job Entry Batch
Self-Contained Interactive
1c
Plotter
Graphics
1d
Keypunch/ Key Entry

Level 2

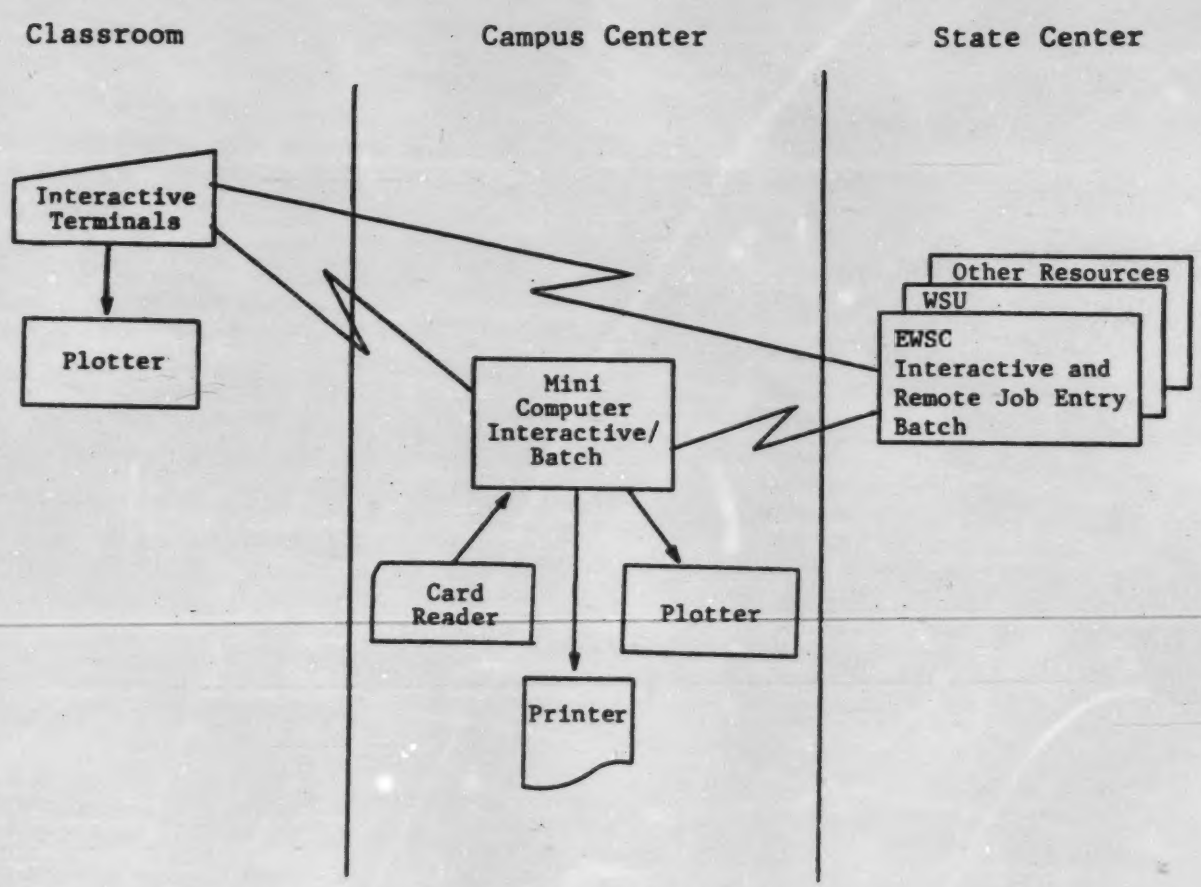
At the second level of the stratified delivery system is the Campus Center. (See figure 3.) The Campus Center provides in addition to Level 1 capabilities both local and remote interactive computing plus local and remote batch RJE computing support.

Interactive terminals may be used with local mini-computers or service centers enabling the user to select services required for each course offering. High volume programming activities in the BASIC language, for example, would be available at low cost on campuses with more sophisticated software and application packages available from appropriate state service centers or other sources.

Level 2 Campus Center

Figure 3

LEVEL 2: Campus



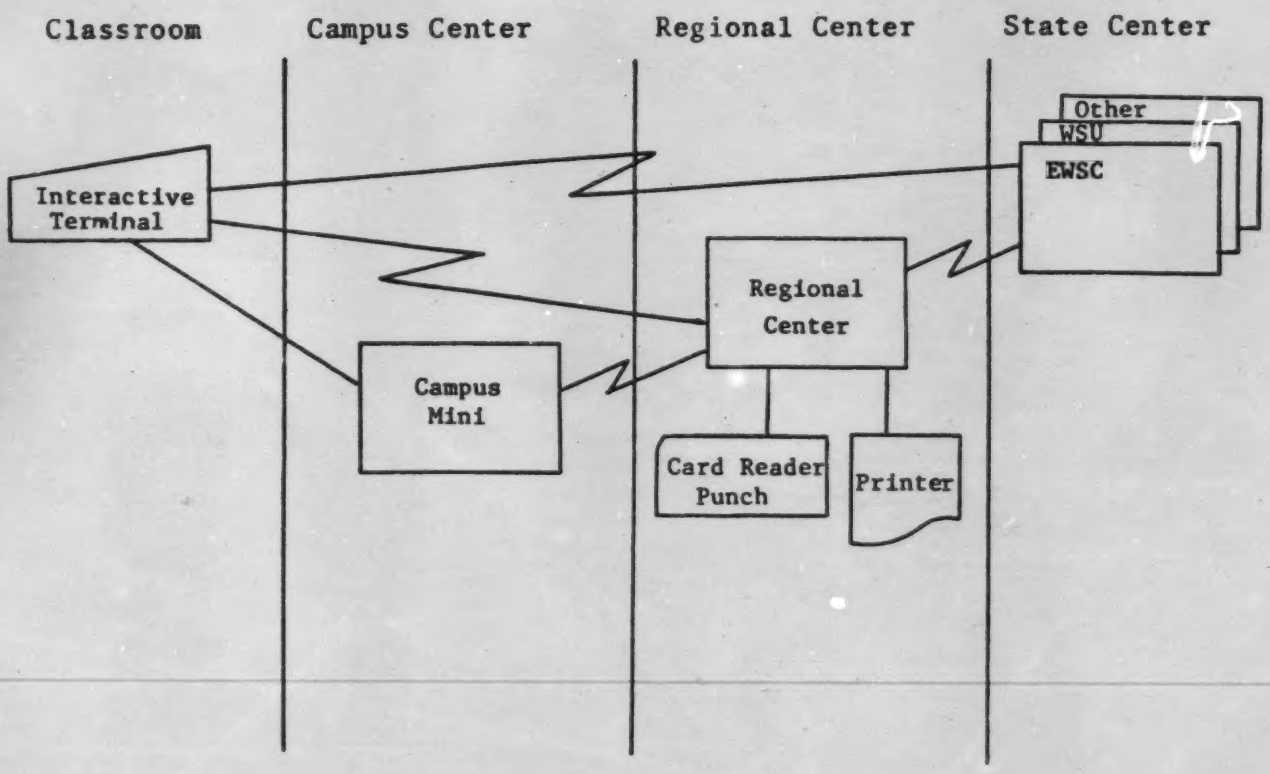
Level 3

At the highest level of the stratified delivery system is the regional center Level 3. The regional center provides Level 2 capabilities for two or more campuses. Similar in configuration to Level 2 facilities Level 3 provides interactive and batch processing local and remote for the involved campuses.

Level 3 Regional Centers

Figure 4

LEVEL 3: Regional



VI. PEOPLEWARE

VI.1 OVERVIEW

Of all the resources necessary to support the curriculum content and curriculum objectives of instructional computing in the community colleges of the state, none is more important than a staff of competant, qualified, and well-trained personnel. Without the presence of such people, the best hardware and the best software may be inefficiently utilized. It is safe to conclude that without adequate committment to a well-trained staff and an ongoing program of personnel development, a college will not achieve maximum benefit and return on its investment in hardware and software. The following sections identify those peopleware resources necessary to support occupational and academic instructional computing activities.

VI.2 PEOPLEWARE - DATA PROCESSING TECHNOLOGY

A versatile, technically competent staff is absolutely essential to a sound community college program in data processing. While providing the faculty with the time and opportunity for growth and skill, the attainment of new skills is critical. Constant study and involvement is necessary if the staff is to avoid stagnation, technical obsolescence, and is to keep abrest of new developments.

SPECIFICATIONS

1. Instructional Qualifications

The ideal data processing faculty member at a community college should have at least two years of programming experience, including demonstrated competence on the computing resource that is being used. For example, assembler language instructors should be competent to write programs in the assembler language. Instructors of high-level languages should know at least two high level languages, and should be aware of special-purpose languages in sufficient detail to be able to discuss types and uses with students. Also, the faculty should be competent in the particular executive or or operating/control system installed on the computing resource in use, and should be able to distinguish functions and capabilities of various other systems for students.

Most community colleges will find it difficult to locate faculty with all of these qualifications. As a result, in-service and pre-service training programs are required to help faculty members acquire and update necessary technical skills to teach data processing.

Specific requirements for vocational instructor selection and certification for the State of Washington may be found in WAC 131-16-091 through WAC-131-16-093 of the Standard Policy and Procedures

Manual of the SBCCE. Local technical advisory committees may be of assistance to the college in the selection and upgrading of faculty. See Section VII of this document for a detailed description of training objectives and resources available.

2. Technical Support

Technical support for the occupational programs may include paid para-professionals and/or student lab assistants. It is recommended that each college have personnel from these two categories to maintain an adequate computing environment.

Colleges using state host facilities must have the support of host center personnel in order to adequately utilize the facilities.

Cooperative work experience programs must have professional staff assigned to this activity on a paid or released-time basis.

Preferably a full-time person should be placed in this position if enrollment levels are sufficient.

VI.3 PEOPLEWARE - ACADEMIC INSTRUCTIONAL COMPUTING

Although academic computing does not require that the instructor be a professional programmer or computer technician, as is the case for the occupational data processing technology curriculum, it is nevertheless essential that academic instructors have experience and knowledge of the applications or languages being taught. Clearly, one cannot effectively teach what one does not know. However, experience and knowledge in computing and programming is at best difficult to obtain for the faculty member faced with incorporating in his classes computing support. In service training, released-time programs, and seminars can go a long way to fill the need for a faculty knowledgeable of the role and technology of computing in academic instruction.

This section defines the peopleware skills and capabilities needed by faculty engaged in teaching courses that utilize computers and computing technology.

General Introduction to Computing Courses: The faculty member teaching a course of this type needs extensive general knowledge of computers, data processing, and automation. This requisite knowledge needs to be broad and somewhat philosophical. Specific technical expertise is not as crucial as a well conceptual knowledge of historical aspects of computers, computer applications, architecture of computational tools, roles and impacts of computing on society, and prospects for the future in computing and automation.

Problem Solving and Language Courses: Here the faculty member should be a qualified person capable of not only teaching the

subject matter area but also the use of the language, languages, or application programs involved. He should have an operational knowledge of the use of the campus computing facilities supporting his course content.

Instructors of language courses shoulc have training in the language, for example, a formal college course in the language, or should have demonstrated competancy evidensed by programs written, seminars attended, or other information.

CAI, CMI, CBI: Peopleware support for CAI requires that the instructor be able to develop new or utilize existing CAI software/hardware packages.

To be able to use, prepare and/or modify CAI material, individual instructors must be trained in the authoring language available to them. This training will require seminars be held and may require release time for the instruction of the language.

Technical staff support must be available both at the campus center and at the host computer center to handle problems that develop.

Peopleware Support Required to utilize an existing CMI system implemented on an existing state host includes a knowledgeable instructor familiar with terminal operation and access to host facilities is necessary, assuming the instructor already knows the characteristics of the CMI package selected. To develop a CMI package, programmers, KP - clerical personnel, and an instructor for the subject matter to be supported by the CMI system so-developed should be available.

CBI, like problem solving and language areas, requires an instructor with knowledge and skills in utilizing CBI applications. This involves experience in the use of computing resources, familiarity with software support programs, and competancy in applying CBI applications to courses being taught. An instructor should have some prior experience in using CBI applications before engaging in such activities in the classroom.

Computer Science: The computer science instructor, like his data processing technology counterpart, should be a computing professional, with recognized academic credentials.

VII OTHER REQUIREMENTS

VII.1 OVERVIEW

Other areas of need in meeting instructional computing requirements are briefly discussed in this section. Areas covered include personnel development, instructional aids, and facilities.

VII.2 PERSONNEL DEVELOPMENT

Continuing on-the-job training is essential for all instructional staff members. Data Processing Technology Instructors must be aware and knowledgeable of new developments in computing technology and applications in order to make judgements as to how, when, and where these new concepts are to be integrated into curriculum. Theoretical knowledge and practical experience of both equipment and techniques is required. Academic instructors, too, need exposure to new developments and techniques in their areas of instructional computing.

Major sources of knowledge on programming techniques and applications are professional journals, seminars, conferences, and in-service training programs. Other sources of information are specialized summer institutes, senior college programs, and manufacturer or vendor schools. An implication to consider is the fact that instructional personnel in computer technology have very few senior institutions in which to pursue additional training in certain required skills. Although systems training and other skills in computing science are offered, these areas are not geared to the level of instruction in community college programs. Consideration therefore, must be given to released time for instructors to obtain training in vendor schools, technical seminars, independent study, or on-the-job experience.

IN SERVICE EDUCATION

In-service training for faculty members is an essential component for on-going success of instructional computing. Reference here is not simply to faculty members who are directly involved with data processing technology instruction; or even simply to faculty members in disciplines which have traditionally demonstrated the highest interest level in the use of the computer in instruction, i.e., science and mathematics. Colleges must be willing to spend resources and to plan carefully for the education of faculty members in all disciplines and occupational programs in order to foster an awareness of what the computer can do, how it works, what it cannot do, how it is misused, and how to use this technological development in a human environment. There is no learning experience which could not, in some conceivable way, be assisted by a computer. (Not all such conceptions would be practically feasible. Nevertheless, the range of practical applications considerably exceeds those permitted by stereotypes and

inhibitions which tend now to limit faculty thinking about the use of the computer.)

The plan for in-service training should include sufficient support and hardware capabilities to allow faculty members the time to experiment and "play" with the computer in a direct real-time environment.

There is currently available from the Department of Personnel Interagency Training Division a complete inventory of audio-visual tapes on data processing. The tapes are available at no cost and address a wide variety of subjects such as Computing Systems, Fundamentals, Teleprocessing Concepts, FORTRAN IV, DOS, OS, Data Base Concepts, Simulation, Project Management and many more. A complete, current listing of the courses and how to obtain them can be obtained by writing to the Interagency Training Division, 910 East Fifth, Olympia, WA 98504.

IN-SERVICE TRAINING OBJECTIVES

1. The provision of in-service training faculty and staff, the availability of ideas and help for those interested in using the computer as a teaching-learning tool.

2. The improvement of instructor proficiency through the monitoring student progress data.

3. The conduction of pilot projects involving the use of computers which will enhance the instructional programs.

4. The development of a library of CAI, CMI, CBI and problem-solving applications and documentation.

5. The evaluation of video tape and programmed instruction course usage to assist instructors in teaching problem-solving languages.

6. The conduction of periodic workshops wherein computer-oriented teaching ideas can be exchanged and developed.

7. The development or adoption of Computer-Assisted, Computer-Managed, and Computer-Based instruction module for demonstration purposes.

8. The orientation of instructors to the many different areas in which the computer can be applied to help solve a problem.

9. The development of an in-service educational package that will provide enough experiences for instructors to better understand the comparable problem-solving capabilities and how they might be used in better teaching his/her subject.

10. The provision of college-credit courses for faculty on the effective uses of computer-supported instruction.

11. The provision of staff release time for program improvement.

12. The development of a suggested instructor profile including self-improvement opportunities and desired qualifications.

VII.3 INSTRUCTIONAL AIDS

Various mediums and information resources are desirable in addition to lectures, textbooks and computing equipment. Some of these may be prepared by individual instructors although in many cases professionally prepared materials are available. This section briefly addresses those media.

1. Audio Visual
2. Transparencies
3. Filmstrips, Prepared Type
4. Movie Filme
5. Video Tape
6. 35mm Slides
7. Audio Tape Lectures
8. Synchronized Sound Projector/Viewer Lectures
9. Terminal Lectures or Lessons
10. Microfische
11. Simulators and Simulation Models

VII.4 FACILITIES

Physical planning for facilities consists of providing for adequate student and staff work space plus providing for location of computing equipment. Vendor specifications must be considered when installing equipment for each of the three levels of delivery system.

VIII RESOURCE SPECIFICATION AND ACQUISITION PROCEDURES

VIII.1 INTRODUCTION

In order to support the activities described in the proceeding sections of this document, many of the community colleges will desire new or additional equipment and access to appropriate software support. Computing equipment for a given college should be acquired in a phased growth manner with the amount of computing resources reflecting the experience and growth of instructional computing on campus.

VIII.2 ACQUISITION PROCEDURES

The process of acquiring computing resources consists of five general steps: 1) defining the campus' instructional needs; 2) developing alternatives to satisfy those needs; 3) selecting the best alternative; 4) receiving the necessary approvals, and 5) selection of the equipment.

The first step is to define the campus instructional needs. This involves the completion of the appropriate documents (see Appendix A) - Class Profile (Form 1), Requirements Summary (Form 2), Alternative Cost Analysis (Form 3), and Alternative Cost Summary (Form 4). The appropriate documentation is dependent upon the campus' location within the stratified delivery system. For complete description of the recommended information see the detailed section on each level.

The second step is to determine various alternative methods to meet the defined needs. The alternatives considered will again be dependent upon the campus position within the stratified delivery system.

Step three is the selection of the best alternative. Some of the factors which should be considered are: meeting defined needs, cost, and a means for future growth.

There are essentially three levels of approval which are necessary before ADP equipment is acquired: 1) institutional; 2) State Board for Community College Education; and 3) the Data Processing Authority. Institutional approval includes appropriate expenditure approoval and appropriate instructional program approval.

Institutions may request ADP equipment by submitting the appropriate documentation to the State Board office. The State Board for Community College Education acts as a liaison between the colleges and the Data Processing Authority. Thus any equipment which requires DPA approval will be reviewed by the SBCCE within the framework of community college computing plans and State Board

policies. Approval requests and recommendations will be referred to the DPA.

Under the state ADP Policy PO.02.010 all ADP equipment, except stand-alone devices limited to keyboard or voice entry with output not suitable as an entry vehicle into digital or analog computers (e.g. desk calculator, tape recorder) must be approved by the DPA before they are acquired. The Data Processing Authorigy reviews the individual plans within the context of its approved Data Processing Plans and its existing policies and guidelines.

The Data Processing Authority will also work closely with the requesting campus to aid them in the formal equipment acquisition procedure. The requesting campus may choose to acquire their equipment from an existing state contract; if so the Data Processing Authority will place the order. Otherwise the DPA will work with the campus on a competitive acquisition.

A competitive acquisition involves the development of detailed functional equipment specifications for a Request for Proposal (RFP). This RFP is sent to the vendors. The vendors develop their proposals and submit them to the Data Processing Authorigty. The proposals will then be evaluated by the requesting agency with assistance from the DPA. After the apparent successful vendor is selected, a finalized contract will be negotiated by the DPA with the vendor. Upon completion of the negotiation process the DPA will authorize the requesting institution to proceed with the acquisition.

VIII.3 SOFTWARE ACQUISITION AND MAINTENANCE

Computer software for instructional computing will be provided for the community colleges primarily by host computer center facilities. Access to host software will be via telephone lines and remote terminals. For those campus facilities that are local and/or self-contained, vendor-supplied software may be acquired in accordance with the procedures and forms in the appendixes. Development of software and applications packages may be undertaken at the local campus level for specific instructional objectives. There exist a number of resources within the state - host centers, other four-year and two-year colleges, and various state agencies--to aid colleges in the development and maintenance of necessary software.

However, coordination of available and existent software and applications packages will encourage the most efficient and economical use of these resources. Currently, there is no body, committee, or commission charged with the coordination, dissemination, acquisition, and maintenance of community college software. It is a recommendation of this plan that some form of coordination body be established to perform the afore mentioned tasks. Otherwise, software acquisition, development, and

maintenance will necessarily remain at the campus level where visibility may be low, redundancy high, and costs to the state less than optimum.

VIII.4 PEOPLEWARE ACQUISITION

Acquisition of personnel is a local campus responsibility and should be consistent with the afore mentioned technical qualifications and competancies. However, the coordinating body mentioned in the previous section could be instrumental in establishing recommendations, guidelines, and practices in the area of personnel development.

VIII.5 CRITERIA FOR ENTRY INTO AND MOVEMENT WITHIN THE STRATIFIED DELIVERY STRUCTURE

Initially, a college will enter the instructional computing environment at level 1. The following criteria establishes the steps and requirements to move from level 1 to level 2 and from level 2 to level 3.

Entry Level Criteria

Level 1A--Interactive terminal to remote service center.

1. Desire/interest by instructor for use of computing capabilities in one or more instructional areas expressed by class profile (see appendix A) and letter to State Board for Community College Education. Letter should include designation of host site, dollar cost, and source of funds.

2. Demonstrated Instructional Computing Resource Requirements Analysis to include. (See appendix A)

   a) Class Profiles to identify need for computing support for each instructional program.

   b) Requirements summary which show current level of computing activity and projected use over four years.

   c) Alternative Cost Analysis to show detail current and four year projected costs associated with each alternative.

   d) Alternative cost summary to show comparison of total costs for each alternative on a prorated basis over four years.

3. Sufficient level of planned processing over a four year period to justify selected alternative.

4. Budget commitment and approval by local administration.

5. Approval by State Board and the Data Processing Authority.

Level 1D. Data Entry Stations

(Same as level 1A)

Level 2--Campus Center

Criteria

1. Demonstrated institutional and staff experience in Level 1 type computing activities.

2. Hardware compatability with state centers, regional centers, and other campus sites.

3. Software compatibility with state centers, regional centers, and other campus sites.

4. Identified staff support by the institution for the campus center--(recommended one full-time staff as minimum).

5. Instructional computing resource requirements analysis.

6. Sufficient level of planned processing over a four year period to justify selected alternative.

7. Budget commitment and approval by local administration.

8. Approval by State Board and the Data Processing Authority is required.

Level 3--Regional Centers

Criteria

1. The primary basis for establishing Regional Centers will be improved level of service and cost savings.

2. Institutions wishing to establish a regional instructional computing center should use DPA Standards for Service Center Operations as guidelines.

3. A formal cooperative agreement must be agreed to by all participating schools in the Regional Center.

4. Hardware and software compatability with state Service Centers and other Regional Centers will be required.

5. An Instructional Computing Resource Requirements Analysis will be required.

6. Budget commitments and approval by local districts will be required.

7. Approval by State Board and the DPA will be required.

APPENDIX A

Class Profile

(Example)

1. Course Title: Math 115 Introduction to Computer Programmi
2. Course Objecitve: Programming and coding of problems for digital computers; flowcharts, loops and subroutines.
3. Number of quarters per year course offered: Four Quarters
4. Current Computing Facilities: The Evergreen State College
5. Course Content:

STUDENT ASSIGNMENT PROFILE
(This is an example of one possible course.)

Software	Prog. Assgnt.	Description	Mode	Run 1	Run 2	Run 3	Run 4	Run 5	Time/ Minutes
BASIC	1	Hardware Familar-ization	Batch	100% 5"/S*	50% 5"/S	17% 5"/S			
BASIC	2	Loop & Counters	Batch	100% 10"/S	90% 10"/S	50% 15"/S	17% 15"/S		
BASIC	3	Automatic Looping Print Format	Batch	100% 10"/S	90% 10"/S	50% 15"/S	17% 15"/S		
BASIC	4	List Dimension	Batch	100% 10"/S	90% 10"/S	50% 15"/S	17% 15"/S		
BASIC	5	Two Dimension Array Sub-routines	Batch	100% 10"/S	100% 15"/S	67% 20"/S	33% 20"/S	17% 25"/S	
FORTRAN	6		Batch	100% 20"/S	100% 20"/S	33% 20"/S	17% 20"/S		

*Five minutes per student

	(History)			(Projected)				
	Year 1	Year 2	Current	Year 1	Year 2	Year 3	Year 4	4-Yr. Total
No. of students	82	90	90	90	90	90	90	360

No. of Hours

(Current year amount determined by profile above. Projected based upon current year data.)

Document 2

REQUIREMENTS SUMMARY
(Processing Hours)

	(History)			(Projected)					
Classes	Year 1	Year 2	Current	Year 1	Year 2	Year 3	Year 4	4-Yr. Total	
Totals from class Profiles									
Math									
DP									
Acctg.									
Sub-Total									
Non-Class Needs									
In Serv. Trg.									
Total No. of Processing Hrs.									
Functional Requirements - Software									% Dist.
BASIC									
FORTRAN									
Total									100%
Non-Instructional Requirements									
Admin.									
FINAL TOTAL IN NO. OF HRS.									

Document 3

ALTERNATIVE COST SUMMARY

(All amounts are for a four-year period)

CURRENT OPERATING PROCEDURE

Terminals (off-line)	($)	
Terminals (on-line)	( )	
Time Sharing	( )	
Supplies	( )	
Telephone Lines	( )	
TOTAL OPERATING COST		($)

ALTERNATIVE I (with enhancements over existing)

Total Cost of Alternative		($)
Residual Value:		
Equipment to be amortized:		
a. Computer	($)	
b. Terminals	( )	
c. Communications	( )	
	( )	
25% Residual Value		( )
TOTAL NET COST		($)

Advantages

Disadvantages

ALTERNATIVE II

(Repeat same information as for ALTERNATIVE I)

Document 4

ALTERNATE COST ANALYSIS

COST ITEM	ONE TIME (Purchased)	RE-OCCURING (Operating) Year 1	Year 2	Year 3	Year 4	TOTAL 4-YR. COST
Computer						
Mainframe	($)					($)
Printer	( )					( )
Reader	( )					( )
Maintenance	( )	($)	($)	($)	($)	( )
Terminals						
Terminal 1	( )					( )
Terminal 2	( )					( )
Maintenance		( )	( )	( )	( )	( )
Communications						
Lines		( )	( )	( )	( )	( )
Interface	( )					( )
Coupler	( )					( )
Portable Phone	( )					( )
Other Equipment						
IBM Key Punch	( )					( )
Software						
FORTRAN	( )					( )
Time Sharing Services						
EWSC		( )	( )	( )	( )	( )
Support						
Staff						
Facilities						
Supplies		( )	( )	( )	( )	( )
FINAL TOTAL	($)	($)	($)	($)	($)	($)

APPENDIX B

GLOSSARY OF TERMS

Batch Processing: An approach to processing data where a number of input items are grouped for processing during the same machine run. Contrasted with on-line processing.

Computer Generations: Refers to the technological changes in manufacturing computer equipment which result in new models of machines with increased capacity, new capabilities, and generally lower cost for volume of work produced.

Computer Program: A set of logically arranged instructions to cause the computer to perform its various functions of data input, processing, and data output.

Computer Programing Language: A symbolic or English language designed to communicate problem instructions to the computer. Common languages include COBOL, FORTRAN, PL 1, and Assembler language.

Computer Systems: A network of interconnected machines consisting of a central processing unit, data input units such as card readers, data storage devices such as magnetic tape units, and data output devices such as printers.

Data Files: A collection of related items of data treated as a unit. Example: A student ID number is an item of data as is a course grade, home address, etc. Collectively, these items form a record for each student - the individual records of all students forms a data file.

Data Processing System: A network of machines and procedures capable of accepting data, processing it according to a plan, and producing the desired information results.

Hardware: The physical machines which comprise a computer system. Contrasted with software.

Information Science: A field of study devoted to applying computers to the problems of information needs, collecting and processing information, and preparing information for decision making. Typically includes advanced techniques of operations research, computer modeling and simulation of business processes, and numerical analysis.

Information Systems: A combination of data files, computer programs, and procedures to collect, process, store, retrieve, and disseminate various operating reports and analysis for use by administrators, boards of education, and teaching faculty.

Numerical Analysis: The study of methods of obtaining quantitative solutions to problems in which the elements of the problem can be expressed in mathematical relationships.

Numerical Control: A field of computer applications which center around the control of machine tools by computer programing.

On-Line Processing: A system (using remote terminals) in which data is sent to the computer and/or retrieved at the time and place most advantageous to the user. Contrasted with batch processing.

Remote Terminals: Devices for data input and retrieval to and from a computer. A device is located at some distance from the computer system and may be connected to the computer via a telephone communications systems or by direct cable.

Software: The computer programs are prepared by either the manufacturer or the user to control the functions of a computer system. Contrasted with hardware.

System Analysis: The study and analysis of an administrative activity to determine the processes, procedures, sources of data, and elements of information that are necessary to accomplish the purpose of the activity.

Peopleware: The staff and personnel necessary to operate the hardware, software, and instructional systems used in instruction.

Time-sharing: The simultaneous, multiple use of a computer device by two or more users, usually via terminals in an on-line teleprocessing environment.

Terminal: Any of a wide variety of typically key-driven computing devices that are connected to a remote computer by telephone lines.

Host Computer: Any of a wide variety of computing devices capable of providing computational and data processing support to remote terminals.

Programmable Terminal: A product of recent developments in computer terminal technology that provides both off-line and on-line computing activities through the same terminal-type device. Usually, programmable terminals can be used in a wider variety of applications and with greater versatility than standard, typical computer terminals.

Off-Line Computing: The use of stand-alone, dedicated computing resources that are not connected to a host computer through telecommunication lines.

Computer Simulation Model: A mathematical model of a real system or environment that is programmed for and operated on a computer in

order to simulate the real system or environment so that it may be studied and analyzed.

BIBLIOGRAPHY

1. Hill, James and Sedrel, Roy, The Computer and the Junior College, American Association of Junior Colleges, 1969.

2. Gerard, R.W., Shaping the Mind: Computers in Education. In R.C. Atkinson and H.A. Wilson (Eds.) *Computer Assisted Instruction.* New York: Academic Press, 1969, pp. 15-40.

3. Suppes, P., Computer Technology and the Future of Education. In V.M. Howes (Ed.) *Individualizing Instruction in Science and Mathematics.* London: Macmillian, 1969, pp. 129-138.

4. Brightman, Richard W., The Computer and the Junior College: Curriculum, American Association of Junior Colleges, 1970.

5. State of Washington, State Board for Community College Education, A Plan for Community College Computer Development, January, 1971.

6. State of Washington, Coordinating Council for Occupational Education, Occupational Analysis for: Business Data Processing Technology, June, 1972.

7. State of Washington, State Board for Community College Education, A Plan for Community College Computer Development, September, 1972.

8. Prepared for the Council of Presidents, Study on the Feasibility of an Educational Computer Network in the State of Washington, Volume 1, 2, and 3, January, 1973.

9. Washington State Data Processing Authority, Plan for Community College Computing, April, 1974.

10. Weeg, G.P., Regional Networks - Revolution in College Computing, in a speech given at the Second Annual Symposium Pacific Northwestern Cooperative Computing Center, April 13, 1973.

11. Washington State Data Processing Authority, Evaluation of a Study on the Feasibility of an Education Computer Network, February, 1974.

12. Proceedings of the Conference on Computers in the Undergraduate Curricula, Volumes 1 through 6, 1970 - 1975.

13. Harding, L.G.; Salop, P.A.; and Johnson, K.A., An Evaluation of the Use of Chemically Treated Answer Sheets, San Diego, Navy Personnel Research and Development Center, Tech. Report 74-9, March, 1974.

14. Vaught, D.M., The Journal. *At MSU: One Computer for All Applications,* June, 1974, pp. 14-15 and 32.

15. Mayo, G.D., Journal of Educational Technology Systems, *Computer Based Instructional Systems*, Winter, 1974, pp. 191-200.

16. Hansen, D.N., The Role of Computers in Education During the '70's, Tallahassee, Florida State University, CAI Center, Tech. Memo No. 15, May, 1970.

17. Hagerty, N.K., Development and Implementation of a Computer-Managed Instruction System in Graduate Training, Tallahassee, Florida State University, CAI Center, Tech. Report No. 11 June, 1970.

18. Hobson, E.N., Empirical Development of a Computer-Managed Instruction System for the Florida State University Model for the Preparation of Elementary School Teachers, Tallahassee, Florida State University, CAI Center, Tech. Report No. 8, April, 1970.

19. Lawler, R.M., An Investigation of Selected Instructional Strategies in an Undergraduate Computer-Managed Instruction Course, Tallahassee, Florida State University, CAI Center Tech. Report No. 19, April, 1971.

20. Rockway, M.R.; and Yasutake, J.Y., Journal of Educational Technology Systems, *The Evolution of the Air Force Advance Instructional System*, Winter, 1974. pp. 29-39.

21. Kelley, A.C., The American Economic Review, *Tips and Technical Change in Classroom Instruction*, May 1972, pp. 422-428.

22. Bylinsky, Gene, Here Comes the Second Computer Revolution, *Fortune*, November, 1975, pp.134.

23. Watkins, Lloyd, Computers on Campus: Three Reports on What They're Doing, What They Could Do, *College and University Business*, May, 1970, p. 71.

24. Bueschel, Richard, The Dual Role of Timesharing in Education, *Modern Data Systems Journal*, March, 1970, p. 78.

25. Stansfield, David, The Computer and Education, *Educational Technology*, May, 1968.

26. Sloan, Harold S., Costs of Automated Instruction In Higher Education, *Educational Technology*, 1969.

27. Marrow, James; and Norman, Adrian, The Computerized Society, Prentice-Hall, Englewood-Cliffs, New Jersey.: 1970.

28. Howard, Alan, An Analysis of the Effects of Computer-Assisted Repeatable Examinations on Different Student Groups, Unpublished Doctoral Dissertation, University of Washington, 1975.

29. Bushnell, D.D., The Role of the Computer in Future Instructional Systems, *Visual Computer Review*, Supplemental 7, 11 (2), 1963.

30. Caffrey, J.; and Mossman, C.J., Computers on Campus, Washington: American Council on Education, 1967.

31. Evans, N.D.; and Neagley, R.L., Planning and Developing Innovative Community Colleges, Englewood-Cliffs: Prentice-Hall, 1973.

32. Gerard, Ralph W., Computers and Education, New York: McGraw-Hill, 1967.

33. Hoffman, J., Computer Assisted Instruction Revised, *College Management*, October, 1973.

34. Keller, F.S., Goodbye Teacher . . ., *Journal of Applied Behavioral Analysis*, 1968, Volume 1, pp. 79-89.

35. Lippey, Gerald, The Computer Can Support Test Construction In a Variety of Ways, *Educational Technology*, Volume XIII, No. 3, 1973.

36. Lippey, Gerald, Computer-Assisted Test Construction, Educational Technology Publications, Englewood-Cliffs: 1974.

37. Report of the President's Science Advisory Committee, Computers in Higher Education, John R. Pierce, Chairman, Washington, D.C.: 1967.

38. Suppes, P., The Use of Computers in Education, *Scientific American*, 1966.

39. Bushnell, D.D.; and Allen, D.W., (Eds.), The Computer in American Education, New York: John Wiley, 1967.

40. Zinn, K., Computer Assistance for Instruction, *Automated Education Newletter*, October, 1965.

41. Zinn, K., Survey of Materials Prepared for Instruction of Instruction Research via On-Line Computer Systems, *Automated Education Handbook*, E. Goodman (Ed.) Detroit: Automated Education Center, 1965.

42. Ban, Avron; Beard, Marian; and Atkinson, Ricard, A Rationale and Description of a CAI Program to Teach the BASIC Programming Language, *Instructional Science*, Volume 4, No. 1, April, 1975, Elsevier, Amsterdam, The Netherlands.

43. Atkinson, Richard C.; and Wilson, H.A., (Eds) Computer-Assisted Instruction, A Book of Readings, New York: Academic Year, 1969.

44. Sanders, Ray W., Networking, *Datamation*, March, 1974, p. 54.

45. Emshoff, James R.; and Sisson, Roger, Design and Use of Computer Simulation Models, Macmillan, 1970.

46. Eurich, Alvin C., Reforming American Education, Harper-Row, 1969.

47. Weeg, G.P., Trends in Instructional Use of Computers, *On-Line*, Volume 3, No. 2, March, 1974.

48. Meyer, Donald P., Community College Has No Computer, But Look What It Is Doing With CRTs, *College and University Business*, April, 1973.

49. Rockart, John; and Morton, Michael S., Computers and the Learning Process In Higher Education, New York: McGraw-Hill, 1975.

50. Milner, Stuart; and Wildberger, A.M., How Should Computers Be Used In Learning?, *Journal of Computer-Based Instruction*, Volume 1, No. 1, August, 1974, p. 7.

51. Seidel, Robert; Hunter, B.; Kostner, C.; and ubin, M., Obstacles to Widespread Use of Computer-Based Curricular Innovations: A Review, *Journal of Computer-Based Instruction*, Volume 1, No. 1, August, 1974, p. 28.

52. otts, J.S., The Place of Computer Graphics in the Business Community, *Data Management*, September, 1975, p. 44.

53. Kamp, Martin, and Starkweather, John A., A Return to a Dedicated Machine for Computer-Assisted Instruction, *Comput. Biol. Med.*, Pergamon Press, Volume 3, 1973, pp. 293-298.

54. Naylor, J.H.; Bolintfy, J.L.; Chy, K.; and Burdick, D.S., Computer Simulation Techniques, New York: John Wiley, 1966.

55. Ford, John D. Jr., and Slough, Dewey A., Development and Evaluation of Computer Assisted Instruction for Navy Electronics Training, *Research Report SRR70-32*, San Diego: Naval Personnel and Training Research Lab, May, 1970.

56. Homeyer, Fred C., Development and Evaluation of an Automated Language Teacher, University of Texas, 1970.

57. Knuth, D.E., Computer Science and Its Relationship to Mathematics, *American Math Monthly*, Volume 81, No. 4, pp. 323-342, 1974.

58. Bunderson, C. Victor, Justifying Computer-Assisted Instruction and Mainline Instruction, proceedings of the Conference on Computers In the Undergraduate Curriculum, 1970, University of Iowa, pp. 12.5-12.15.

59. Hallen, T.T.; Dunham, J.E.; and Bunderson, C.V., Computer Science Lab Exercises - Qualitative Chemical Analysis, *Science Education*, February, 1968.

UNIVERSITY OF CALIF.
LOS ANGELES

APR 2 1976

CLEARINGHOUSE FOR
JUNIOR COLLEGES